Python Programming for Finance

Introduction

Juan F. Imbet

Department of Finance
Université Paris-Dauphine

September 7, 2021

Content

- **8/9/2021** Introduction and Basic Programming: History, Variables, Control Flow, Conditional Evaluation, Arrays, Functions, Generators.
- **15/9/2021** Intermediate Programming: Numerical Python (numpy), Linear Algebra
- **22/9/2021** Data Analysis: Dataframes (pandas), cleaning and processing data through a Financial Example.
- **6/10/2021** Portfolio Management I: Statistics and Simulation
- **13/10/2021** Portfolio Management II: Optimization and Heuristics
- **20/10/2021** Web Scrapping: How to automatize data collection online through a Financial Example.
- **27/10/2021** Data Distribution: How to build a functional API to distribute data
- **3/11/2021** A grasp of advanced Python usage: Coding Standards, General User Interfaces, Extending Python's Functionality with C++

Course Materials

- The course material, slides and code, will be available on the Github repository of the course, and will be updated on a weekly basis.

Grading

- Final Exam 70%
- Individual Project 20%
- 4 take home quizzes - 10 %

What is Python?

Python is a **high level**, **interpreted**, **interactive**, and **multi-paradigm** scripting language.

- **High level**: Python has a strong abstraction from the details of the computer.
- **Interpreted**: There is no need to compile your program before executing it.
- **Interactive**: You can open a Python terminal and directly write your program.
- **Multi-paradigm**: Python supports different programming and implementation paradigms, such as object orientation and imperative, functional, or procedural programming.

History of Python

- First developed by Guido van Rossum in the early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.
- The name was inspired by Monty Python's Flying Circus (sketch comedy from the British comedy troupe).

Why Python?

- Easy-to-learn
- Easy-to-read
- Easy-to-maintain
- Easy-to-scale
- Relatively fast
- Open source

Python files usually end in .py, while other files that depend on Python but are not written fully in Python like Jupyter notebooks end in .ipynb.

Python in Finance

- Corporations often have a *quant* team whose task is to perform quantitative analysis.
- Although many of the *core* programs used by corporations are written in C++ or other robust languages, Python serves as a wrapper to process, analyze, display, and distribute data.
- The numerical libraries in Python makes it comparable to R or Stata in performing Statistics, and to Matlab for linear algebra.
- Helps automatize research tasks and increase productivity.
- Increasing popularity among developers (2020 Stackoverflow survey) JavaScript (69.7%), HTML/CSS (62.4%), SQL (56.9%), Python (41.6%) and Java (38.4%).
- Is free

Setup

How to set-up Python (Homework) 1/3

Download the latest anaconda distribution from `https://docs.anaconda.com/anaconda/install/index.html` based on your operating system, follow the required installation configuration. The default python version in Anaconda might not coincide with the latest stable Python version. To solve this we can create an *environment* which creates a separate Python installation of a specific version. This is useful to run code that is not intended for the latest Python releases.
Open your Anaconda Prompt. Create an environment called env39 running Python 3.9

```
(base) C:\Users\jfimb> conda create -n env39 python=3.9
```

Activate the environment

```
(base) C:\Users\jfimb> conda activate env39
(env39) C:\Users\jfimb>
```

Activate the environment

```
(base) C:\Users\jfimb> conda activate env39
(env39) C:\Users\jfimb>
```

How to set-up Python (Homework) 2/3

You can confirm the version of Python

```
(env39) C:\Users\jfimb> python
Python 3.9.6 (default, Aug 18 2021, 15:44:49) [MSC v.1916 64 bit (AMD64)] :: .
Type "help", "copyright", "credits" or "license" for more information.
>>> exit() # Type exit() to return
```

Allow jupyter to run on this environment

```
(env39) C:\Users\jfimb> conda install -c anaconda ipykernel
(env39) C:\Users\jfimb> python -m ipykernel install --user --name=env39
```

Install the latest version of jupyter

```
(env39) C:\Users\jfimb> pip install jupyter
```

How to set-up Python (Homework) 3/3

Data Analysis in Python tends to be performed using very standard and powerful open source libraries that might or might not come with the Anaconda distribution. To install most libraries in Python it is enough to use the **pip** command showed above. Recall the name of the environment in parenthesis. Pip performs an installation relative to the current environment. To see what libraries you have installed type

```
(env39) C:\Users\jfimb>pip list
```

You can always install libraries on the run, for now install the Data Science trinity, **pandas**, **numpy**, and **Scikit-learn**.

```
(env39) C:\Users\jfimb>pip install pandas
(env39) C:\Users\jfimb>pip install numpy
(env39) C:\Users\jfimb>pip install scikit-learn
```

Launch jupyter

```
(env39) C:\Users\jfimb> jupyter notebook
```

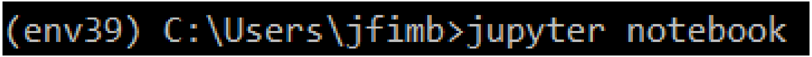
Jupyter

Jupyter is the fastest way to start coding in Python, during the course we will use a hybrid format in which some code will be written in Jupyter and other in standard Python files.

- **Ju**lia **Py**thon and **R**
- The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.
- Open source functionality of high performance languages such as **Mathematica**.

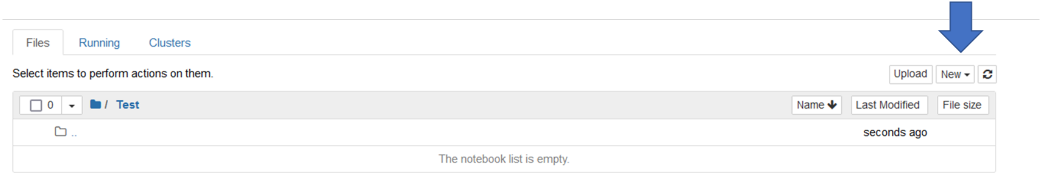
Jupyter

Type **jupyter notebook** in the Anaconda console



```
(env39) C:\Users\jfimb>jupyter notebook
```

Traverse the directory, create a new **notebook** or open an existing one.



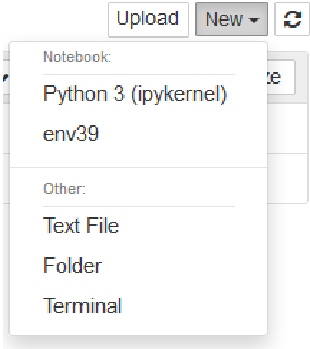
Select items to perform actions on them.

seconds ago

The notebook list is empty.

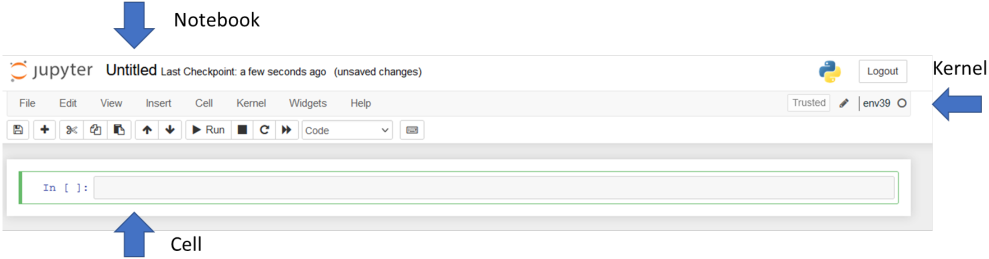
Jupyter

Confirm the **kernel**



Jupyter

Jupyter notebooks `https://jupyter.org/`



Notebook

Kernel

Cell

Jupyter Shortcuts

Shortcuts

- When selecting a cell press **Esc** to get into command mode.
- In command mode press **M** to enter into Markdown mode. Useful for writting text and equations.
- In command mode press **Y** to enter into code mode.
- In command mode press **B** to add a cell below.
- In command mode press **A** to add a cell above.
- Press **Ctrl** + **Enter** to run a cell

Markdown

- Markdown is a lightweight markup language for creating formatted text using a plain-text editor.
- #, ##, ###, for heading.
- $ to encapsulate equations.
- ** For bold text
- * For italics

Git (at home)

- Version control system used to handle software projects.
- Developed by a team lead by Linus Torvalds (creator of Linux).
- We will use it in this course to retrieve and share course material.
- The most used platform that uses Git is Github, if you plan to code for your work get an account!
- Install Git `https://git-scm.com/`
- Open a Github account `https://github.com/`
- Follow the tutorial `https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners`

Python IDE

- An integrated development environment (IDE), helps computer programmers develop software.
- Anaconda provides a ready to use IDE called Spyder.
- An IDE facilitates writing long statements of code, and provide a workspace to follow the evolution of variables.

What is a programming language?

Introduction to programming languages

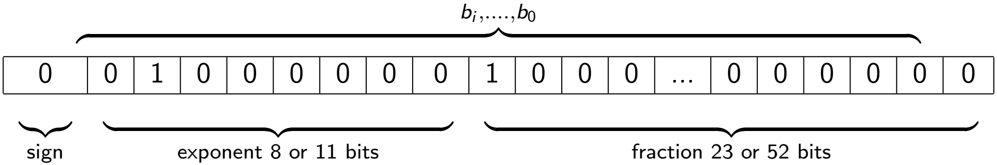
- A high-level programming language provides instructions to perform arithmetic operations in the computer.
- The computer does not understand these instructions, unless they are *translated* to **machine code**.
- This task often requires an intermediary (Assembly language).

Basic computing

- All computer programs can be decomposed into simple and fundamental operations (Bit or logical operations).
- A bit (binary digit) is the simplest unit of information a standard computer handles. It can only hold two values (1 or 0), and is physically represented by a transistor being **on** or **off** in old computers, or by the presence or absence of electrons in semiconductors.
- As bits only take two values, it is natural that computers perform binary arithmetic.

64 and 32 bit architecture

Since computers perform mathematical operations, they require a way to represent numbers. Most computers use up to 64 (32) bits to represent a real number.



$b_i,\ldots,b_0$

0	0	1	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0

sign

exponent 8 or 11 bits

fraction 23 or 52 bits

For 32 bit architecture

$$x = (-1)^{b_{31}} \times 2^{(\sum_{i=0}^{7} b_{23+i}2^i)-127} \times \left(1 + \sum_{i=1}^{23} b_{23-i} \times 2^{-i}\right)$$

Floating point arithmetic

The 64 (32) architecture allows computers to perform floating point arithmetic by representing the whole real line with float numbers.

- Transforms the real line into a **discrete** and **finite** set of numbers.
- Gaps between numbers are not constant, and depend on the magnitude of the number.
- When arithmetic operations result in a number that cannot be expressed exactly as a floating point, the computer approximates the value.
- This can lead to serious numerical problems which will be discussed during the course.