Introduction to Python for Scientific Computing and Data Science

Mohammed Ait Lahcen

Qatar University
Department of Finance and Economics

October 11, 2021

About Python

Modern, high level, free and open source, general purpose programming language.

Used extensively by:

- Tech firms (e.g. Google, Dropbox, Reddit);
- Finance industry (e.g. hedge funds);
- Research agencies (e.g. NASA, CERN);
- Academia.

Why Python?

Python is a free and open source programming language:

- **Free** as in freedom (libre);
- **Free** as in "free food" (gratis).

This means:

- Free to install and use;
- No license issues;
- Source code can be freely read, modified and shared.

Why Python?

- Simple to learn;
- Clean, elegant and very readable syntax;
- High productivity;
- Vast collection of libraries for almost everything;
- Powerful enough for scientific computing;
- Relatively simple tweaks offer performance comparable to compiled languages such as C and Fortran.

Why Python?

Rank	Language	Type	Score
1	Python		100.0
2	Java		95.3
3	C		94.6
4	C++		87.0
5	JavaScript		79.5
6	R		78.6
7	Arduino		73.2
8	Go		73.1
9	Swift		70.5
10	Matlab		68.4
11	Ruby		66.8

Figure: IEEE overall ranking, 2020

Why Python?

Oct 2021	Oct 2020	Change	Programming Language	Ratings	Change
1	3	↑	Python	11.27%	-0.00%
2	1	↓	C	11.16%	-5.79%
3	2	↓	Java	10.46%	-2.11%
4	4		C++	7.50%	+0.57%
5	5		C#	5.26%	+1.10%
6	6		Visual Basic	5.24%	+1.27%
7	7		JavaScript	2.19%	+0.05%
8	10	↑	SQL	2.17%	+0.61%
9	8	↓	PHP	2.10%	+0.01%
10	17	⇈	Assembly language	2.06%	+0.99%

Figure: TIOBE index top 10 languages, October 2021.

Why Python?

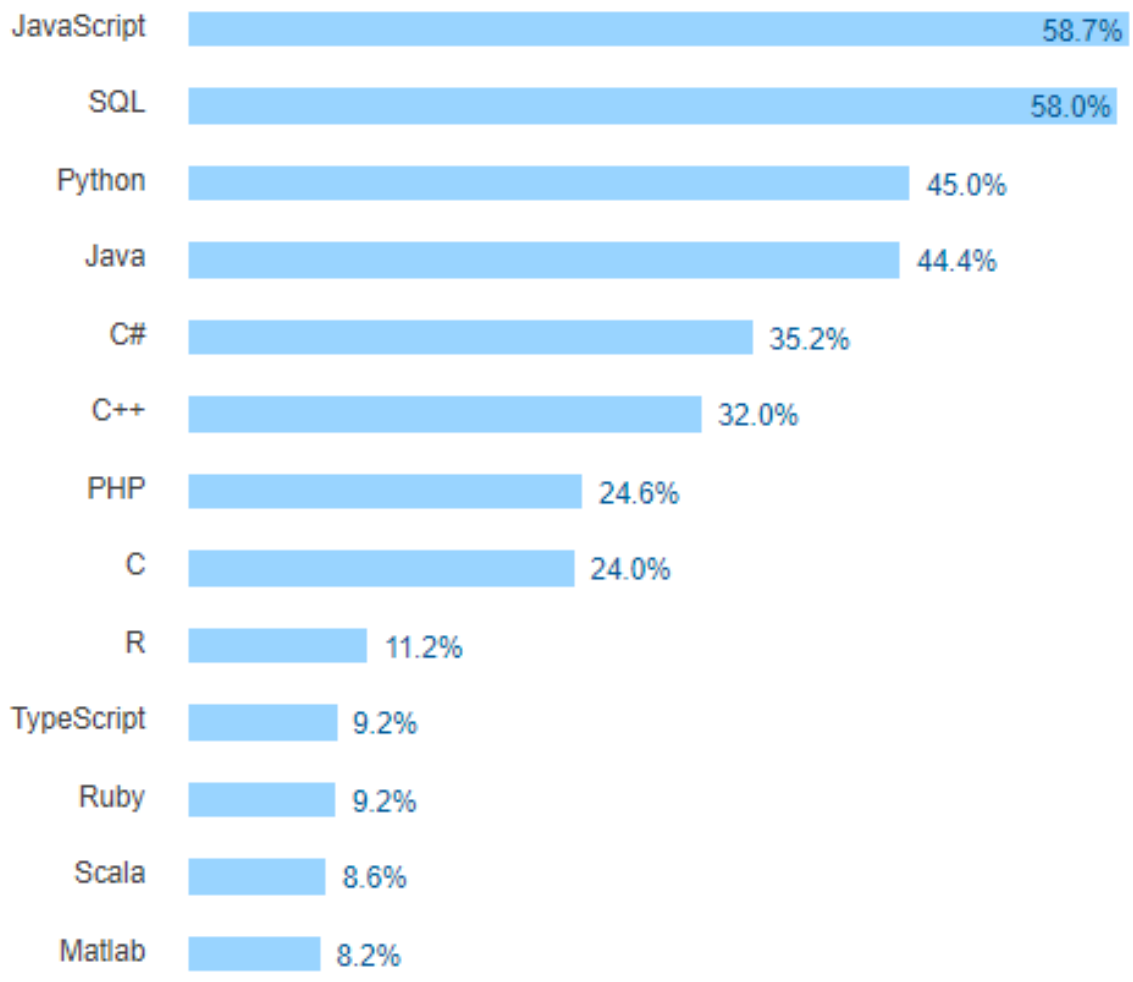

Figure: Most Popular Languages for "Data Scientist/Engineer" occupation. Source: Stack Overflow Survey 2017

Major Scientific Computing and Data Analysis Libraries

- **Numpy**: basic data types, array operations.
- **Scipy**: high-level numerical routines (e.g. integration, interpolation, optimization).
- **Matplotlib**: plotting 2D and 3D figures.
- **Sympy**: symbolic math computations (similar to Maple/Mathematica).
- **Pandas**: data manipulation.
- **Statsmodels**: statistics and econometrics.
- **Scikit-learn**: machine learning.
- **TensorFlow**: machine learning.
- **Numba**: just-in-time compilation for higher performance.

Objectives and Agenda

- Objectives:
  1. Overview of Python.
  2. Some examples.
  3. Discussion.
  4. Resources for further study.

- Agenda:
  1. Core Python: data types and structures, basic operations, input-output, control flow, functions.
  2. Scientific libraries: Numpy, Matplotlib, Scipy.
  3. Data science: Pandas, statsmodels.

Getting started

It is strongly recommended to install one of the many Python distributions (e.g. Anaconda, Canopy, WinPython) and to choose a good programming interface (e.g. Jupyter Lab, VScode, Spyder, PyCharm).

For this class, we will be using:

- Anaconda with Python 3.8;
- Jupyter Lab.

Anaconda

Most popular scientific Python distribution!

Installation:

- Download from `https://www.anaconda.com/products/individual`
- Choose Python 3.8;
- Installation guide available at `https://github.com/maitlahcen/qu_cbe_python_workshop`

Jupyter Lab

For the tutorials, we will use Jupyter Lab:

- Browser based front-end for over 40 programming languages (e.g. Python, R, Julia, C++);
- Allows for live code, equations, visualizations and explanatory text.

Jupyter Lab is included in Anaconda:

- First, install Anaconda
- In the command line type: `jupyter lab`
- Uses Jupyter notebook files with extension `.ipynb`

Resources

- Worshop's Github repo:
  `https://github.com/maitlahcen/qu_cbe_python_workshop`
- QuantEcon Python lectures site:
  `https://quantecon.org/lectures/`
- Scipy lecture notes:
  `http://www.scipy-lectures.org/`
- Scipy cookbook:
  `http://scipy-cookbook.readthedocs.io/`
- Q&A on Reddit:
  `https://www.reddit.com/r/Python/`
- Q&A on Stack Overflow:
  `http://stackoverflow.com/questions/tagged/python`