Christoph Federrath

IDL AND PYTHON IN ASTRONOMY

Interpreted languages in astronomy

For complex data analysis, we want to:

- Interact with our data –> interpreted language
- Be moderately fast, with an option to be very fast.
- Have the ability to write and test complex code
- Have great astronomy libraries.
- Gain skills relevant for employment.

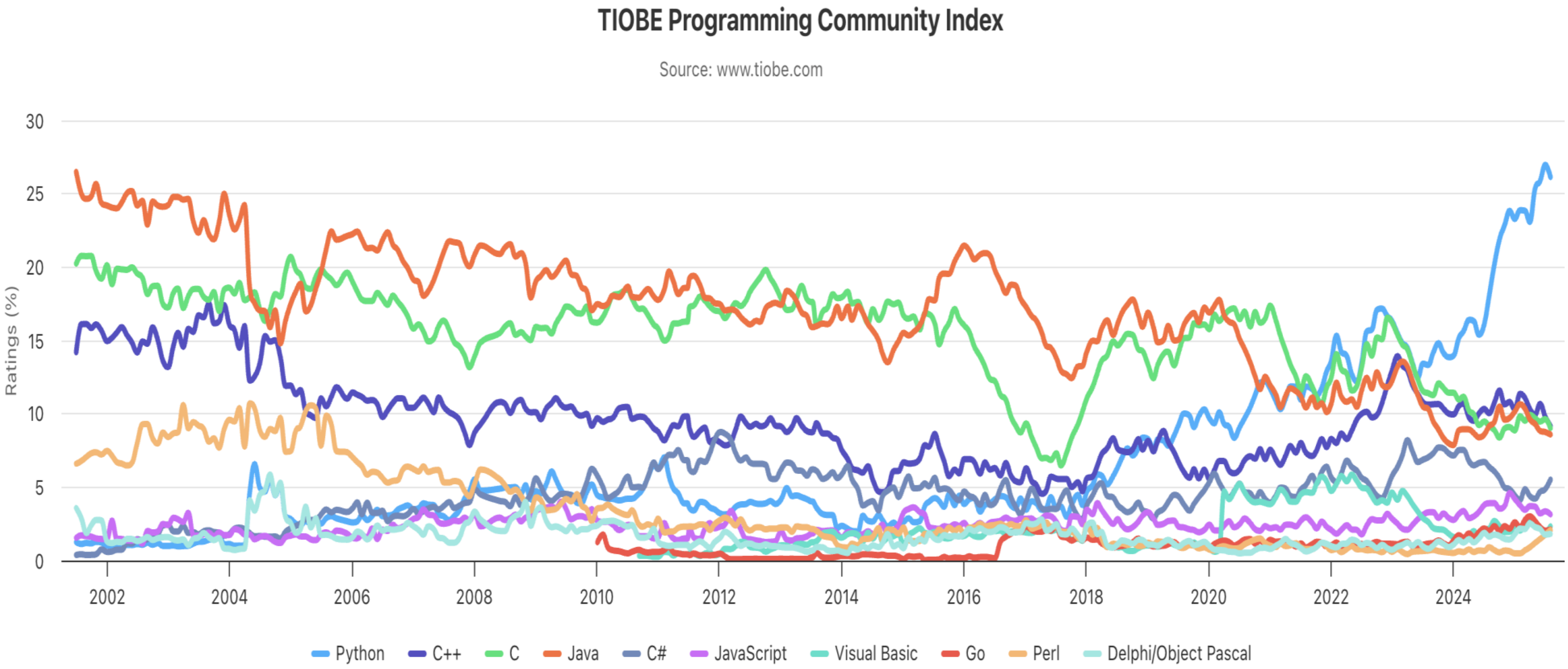
What are the options and their popularity?

- Python (ranked #1)
- R (ranked #14)
- MATLAB (ranked #16)
- Julia (ranked #30)
- IDL (off the list :-)

Rankings from http://www.tiobe.com/tiobe-index/

(note: C++/C are #2/3)

What are the options and their popularity?



TIOBE Programming Community Index

Source: www.tiobe.com

IDL – the good and bad

IDL costs (at lot)

IDL is still in many astro work pipelines and legacy codes running (e.g., HST)

Many current astronomers are still using IDL

IDL is like an “olden-days” version of Python with Fortran style under the hood

Many powerful IDL packages exist

Python – the good and bad

Python is free, with the first version with community support python 2.0 (2000).

Python is *minimalist* in many aspects of the language. There are only a few 10s of keywords in the global namespace (many of which you'd never use).

Python has *lightweight object-oriented* programming. Lightweight because there is no explicit C++ style declarations (no *public, private, virtual, overloaded functions, pointers…)*

Python is designed from the ground up for powerful data structures: lists, dictionaries and tuples (and sets).

However, Python requires external *modules* to have more than basic functionality. If only all useful modules were packaged and available in a neat way…

Use a Python *distribution* or package manager

- Popular for Mac: macports; Windows/Mac/Unix/Linux: anaconda
- *astropy*, which is a great general purpose astronomy package.
- Most additional packages can be installed with *pip,* e.g. *pip install astropy*
- Given that python is open-source, many packages are on github or other public repositories.

Most Important Packages

Numpy: a set of structures and routines designed to make python roughly as powerful as Matlab.

Scipy: Mostly wrappers for powerful libraries such as binning, special functions, etc.

Matplotlib: A plotting package.

Astropy

All of these are on github, and PyPI (pip install…)

We will also use **cfpack**, which contains useful wrappers/demos: pip install cfpack

Some getting started

- If you are not a python expert yet:
  - Learn python for fun, e.g. https://learnpythonthehardway.org/book/