

Matlab vs Python

as tools for psychology and data science

Why raise this question at all?

1. Python is growing really fast for data science
2. Python is free (in every sense)
3. The lab's connection to Matlab is not too strong, so the switch shouldn't be too painful
4. Matlab is not a good language

Problems of Matlab (relevant to the lab)

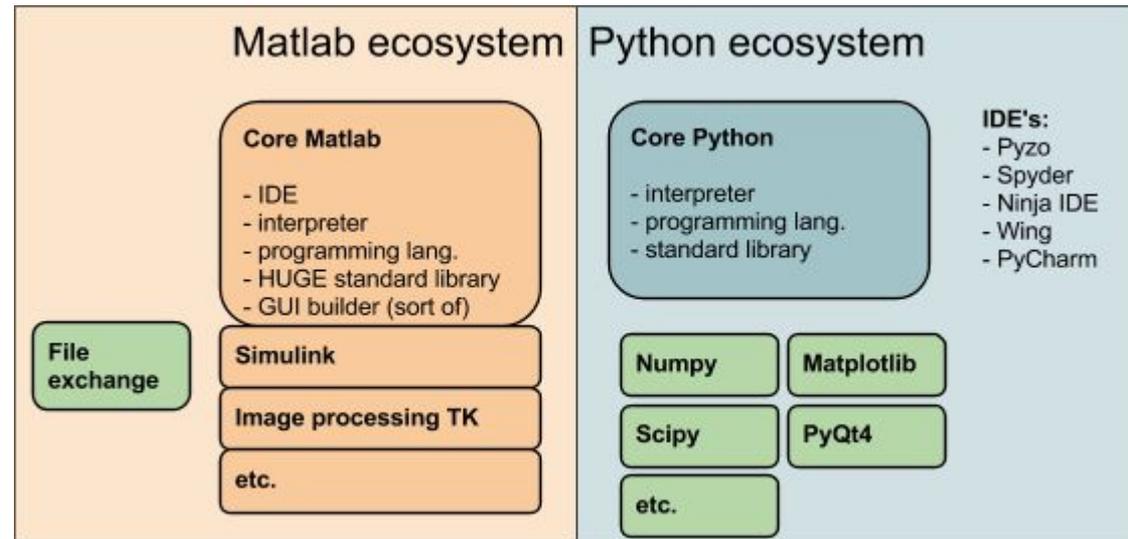
- Not free
- Not as popular for data science as Python
- Poor portability
- Has problems as a language
 - Functions as separate files
 - 1-based indexing
 - Everything is a matrix
 - Big projects use “folders added to current path” approach

What is important?

- Available tools
 - Psychophysics
 - Data science
 - General purpose scripts
- Usability
 - Language
 - Code analysis, profiler, etc (tools that make it easier to write code)
 - Debugger, variable editor (tools that make it easier to find mistakes in program logic)
 - Project

Ecosystems

- Matlab ecosystem consists of a single package with everything + specialized toolboxes like DeeBNet or Psychophysics Toolbox
- Python ecosystem has many modules and you need to combine them yourself into a system that suits you



Comparison (pro-Matlab)

- Matlab is easy for beginners
- There is no need to construct ecosystem from parts - Matlab works out of the box
- Matlab is very popular in psychology
- Matlab has great built-in documentation available at any time
- Matlab has good GUI tools for learning
- Matlab has built-in visualization, variable editor, debugging tools
- Matlab toolboxes are compatible with each other
- Internal functions are fast in Matlab as it is just-in-time compiled

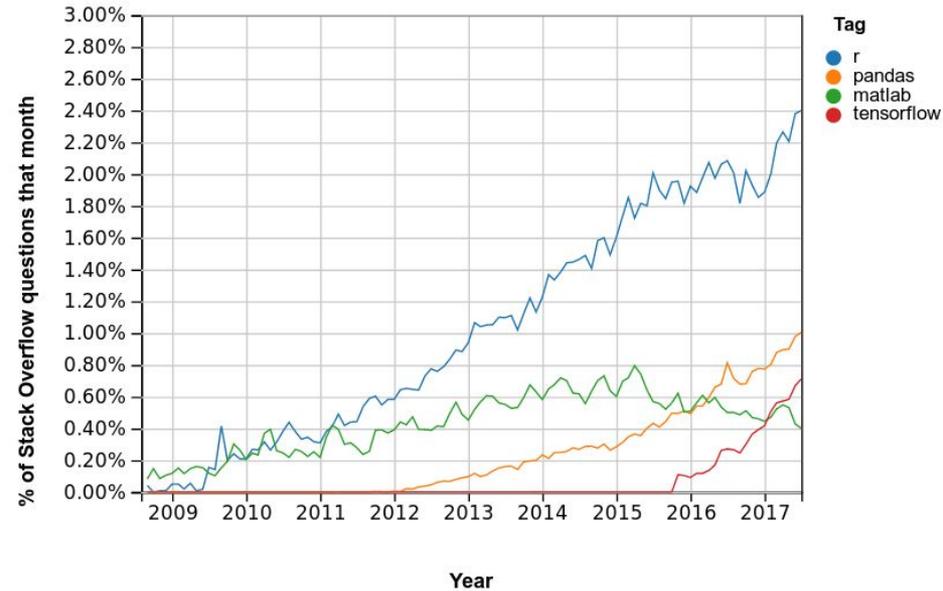
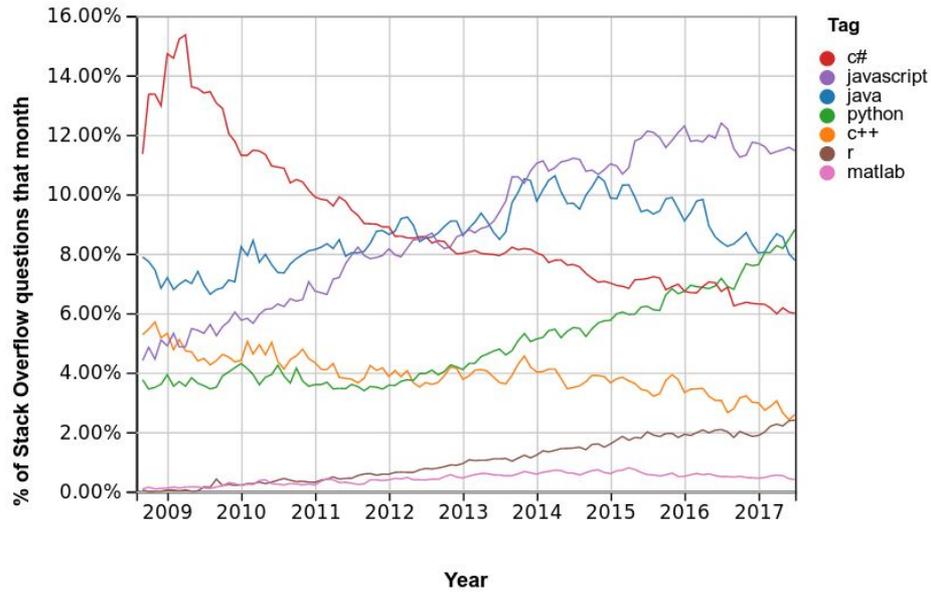
Comparison (pro-Python)

- Python is a good general-purpose language
- Python itself is lightweight
- There is a distribution of Python that is basically “everything works out of the box” (Anaconda)
- There is a very powerful IDE with a free Community license for everyone and a free Professional license for students (PyCharm)
- Python is exploding in popularity in data science and in general
- Python has modules which helps managing bigger projects (compared to “folders in current path” system of Matlab)
- Python has a JIT-compiled implementation (PyPy) which boosts speed
- Numpy (the library for math) internally works in C, so its functions have the speed of C

Comparison (pro-Python)

- Python is very popular in data science and machine learning
 - it is very likely than any deep learning code you find on the internet will be written in Python
- Python has a psychology toolbox - PsychoPy
- Python is free
- It is easier to share Python code (The person you share it with has to either have Matlab, which is very expensive and not very popular as Python in general, or to use Matlab Runtime with precisely the same version as the original Matlab used to compile code)
- Python has great linters that follow PEP8 making code clean and readable (Matlab as a linter as well, but only for functional problems)

Growth of Python



Green on the first graph, orange and red on the second one. Python's growth doesn't seem to slow down or hit a plateau.

Growth of Python

Most of Python's growth is explained by data science field

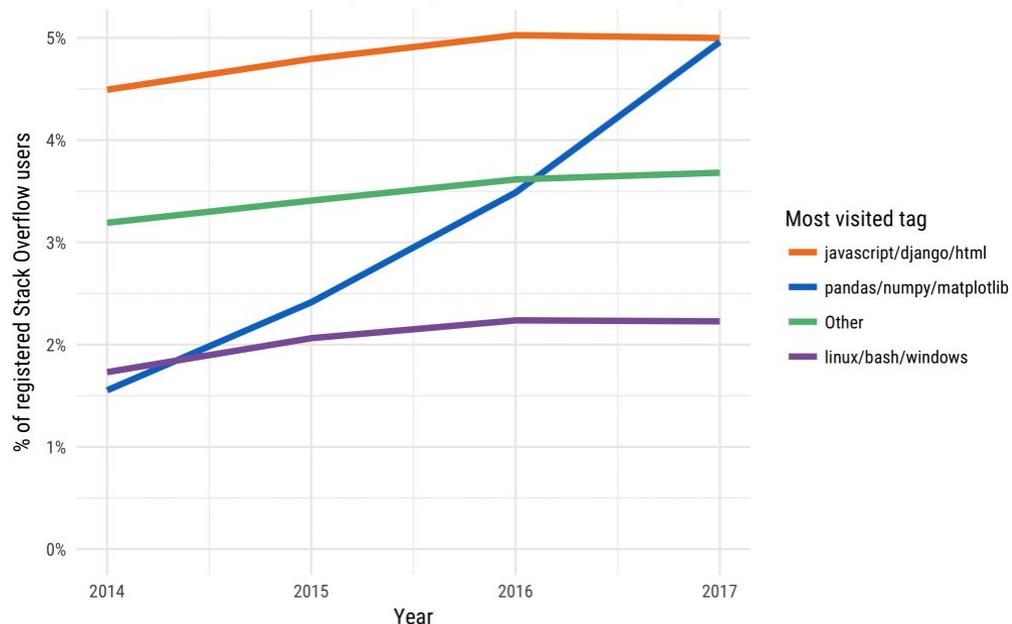
Articles about growth of python:

<https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

<https://stackoverflow.blog/2017/09/14/python-growing-quickly/>

Estimated categories of Python developers over time

Categorizing registered Stack Overflow users, based which of the tags (among those listed) was most visited. 'Other' means none of the listed tags made up more than 5% of the visitor's questions



Stack Overflow Survey

Year	Python popularity	Matlab popularity	Python love (want to stay)	Matlab dread (want to switch)	Want to start using Python
2018	38.8%	5.8%	68% (#3)	77.4% (#6)	25.1% (#1)
2017	32%	4.3%	62.7% (#6)	72.3% (#5)	20.6% (#1)
2016	24.9%	-	62.5% (#9)	72.8% (#3)	13.3% (#4)
2015	23.8%	-	66.6% (#10)	65.6% (#4)	14.8% (#3)

How to set up the Python ecosystem

1. Download and install Anaconda (<https://www.anaconda.com/>)
2. That's it, now you have:
 - a. Python
 - b. Package manager (Anaconda Navigator)
 - c. IDE (Spyder)
 - d. Most libraries and packages you would need
 - e. Jupyter Notebook

How to set up the Python ecosystem

1. For a minimal installation you can just use Python from python.org
 - a. A package manager, pip, comes with it
 - b. You can use any text editor to write code
2. If you want lots of features and nice things in your IDE, use PyCharm.
 - a. It integrates with Anaconda, supports numpy, matplotlib, IPython with array viewers, interactive visualizations and more
 - b. It is a powerful IDE with many tools for debugging, static code analysis, autocompletion, project managing

Final points and personal opinions

- Matlab seems to be very good at specialized mathematical research (for example, digital signal processing)
- Python is great as a general purpose language
- Python is very popular and it's not going to die in the foreseeable future
 - It keeps growing faster and there should be a plateau before it starts losing popularity
- Python has its problems, but it is great, constantly improving and free
- Visual Modelling does not need that great mathematical machine that could be the most important argument in favor of Matlab

Thank you