

2017 QuantEcon Workshops

Economic Modeling with Python and Julia

Introduction

September 2017

Team

- Chase Coleman – NYU
- Victoria Gregory – NYU
- Matthew McKay – QuantEcon
- John Stachurski – ANU
- Natasha Watkins – QuantEcon

Thanks to the Alfred P. Sloan Foundation



Workshop Timeline

1. **9:30 am – 12:00 pm**: Intro to Python and Julia
2. **12:00 pm – 1:00 pm**: Lunch
3. **1:00 pm – 4:00 pm**: Applications

Morning Timeline

1. **9:30-10:30** Introduction and First Steps in Python
 - John Stachurski
2. 10:30-11:00 break
3. **11:00-12:00** Introduction to Julia
 - Chase Coleman

Afternoon Timeline

1. **13:00–13:40** Data Analysis with Python
 - Natasha Watkins
2. 13:40–14:00 break
3. **14:00–14:40** Advanced Data Analysis with Python
 - Matt McKay
4. 14:40–15:00 break
5. **15:00–15:40** Dynamic Programming with Julia
 - Victoria Gregory

Aims / Outcomes / Expectations

Aims =

- Give an overview of the languages
- Show some examples
- Discuss / argue
- Resources for further study

Background — Open Source

Proprietary

- MATLAB
- STATA, etc.

Open Source

- Python
- Julia
- R

closed and stable vs open and fast moving

Background — Language Types

Low level languages give us fine grained control

Example. $1 + 1$ in assembly

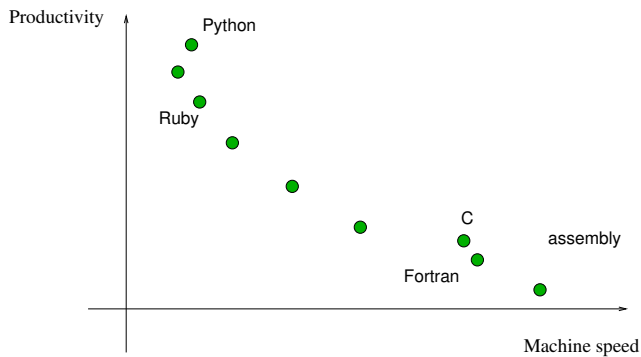
```
pushq    %rbp
movq     %rsp, %rbp
movl     $1, -12(%rbp)
movl     $1, -8(%rbp)
movl     -12(%rbp), %edx
movl     -8(%rbp), %eax
addl     %edx, %eax
movl     %eax, -4(%rbp)
movl     -4(%rbp), %eax
popq     %rbp
```

High level languages give us abstraction, automation, etc.

Example. Reading from a file in Python

```
data_file = open("data.txt")
for line in data_file:
    print(line.capitalize())
data_file.close()
```

Trade-Offs

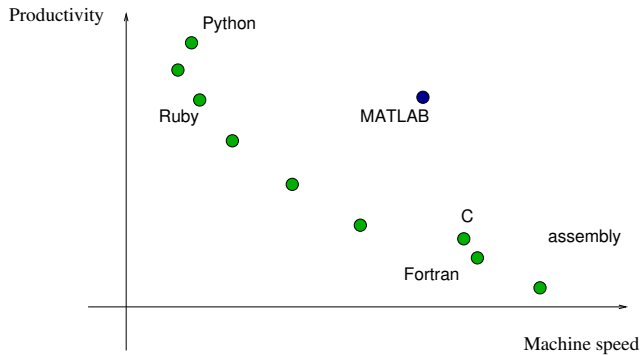


But what about scientific computing?

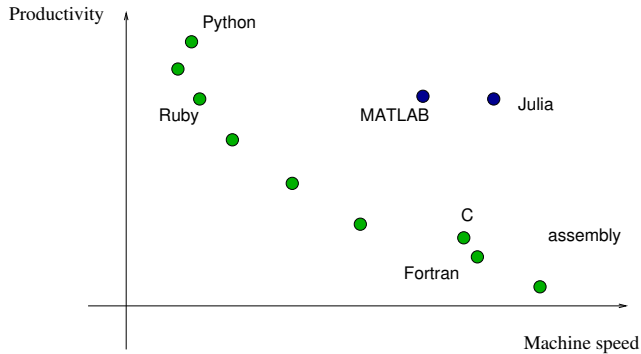
Requirements

- Productive — easy to read, write, debug, explore
- Fast computations

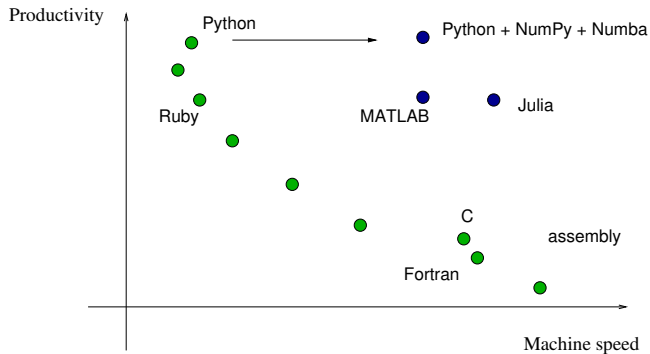
Trade-Offs



Trade-Offs



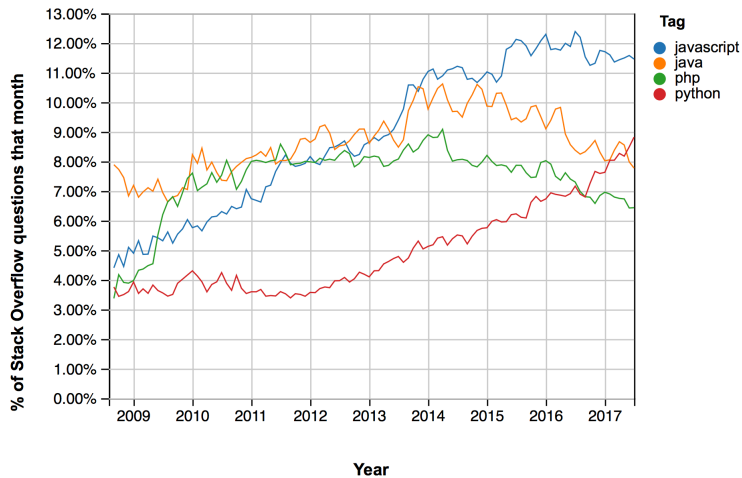
Trade-Offs



Python for Productivity

From local infrastructures to cloud-based systems to building websites to interfacing with SQL databases, Python has nearly limitless applications. Despite its wide-ranging impact, it remains gloriously clean and easy to learn.

– *mashable.com*



Workshop Resources

Cheatsheets, downloads, etc. — see

[https:](https://quantecon.org/2017-phd-workshops-on-computational-methods)

[//quantecon.org/2017-phd-workshops-on-computational-methods](https://quantecon.org/2017-phd-workshops-on-computational-methods)

Download workshop files from the GitHub repo

- via **git** or the **Download** button

Downloads / Installation / Troubleshooting

Install Python + Scientific Libs

- Install Anaconda from <https://www.anaconda.com/downloads>
 - Select Python 3.6
- Not plain vanilla Python

Jupyter notebooks

A browser based interface to Python / Julia / R / etc.

Step 1: Open a terminal

- on Windows, use Anaconda Command Prompt

Step 2: type `jupyter notebook`

Cloud-based server

`workshop.quantecon.org:8000`

Password = **economics**

Topics:

- opening a notebook
- executing code
- edit / command mode
- installing `quantecon`
- getting help
- introspection
- math and rich text

Python for High Performance Computing

- See `John/numba.ipynb`

Exercises

- See `John/plots.ipynb`