QuantEcon: Introduction to Python

GC CUNY

Lecture 1

*Much of the original content in these slides was written by John Stachurski — Many thanks to him for sharing these materials and for generating such great content!

Set up and Resources

Follow the instructions here

- https://github.com/QuantEcon/GC_CUNY_workshop_2019

Assumptions / Prerequisites

- Coding experience is assumed
- But no Python required

Programming Background — Software

A common classification:

- low level languages (assembly, C, Fortran)
- high level languages (Python, Ruby, Haskell)

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.

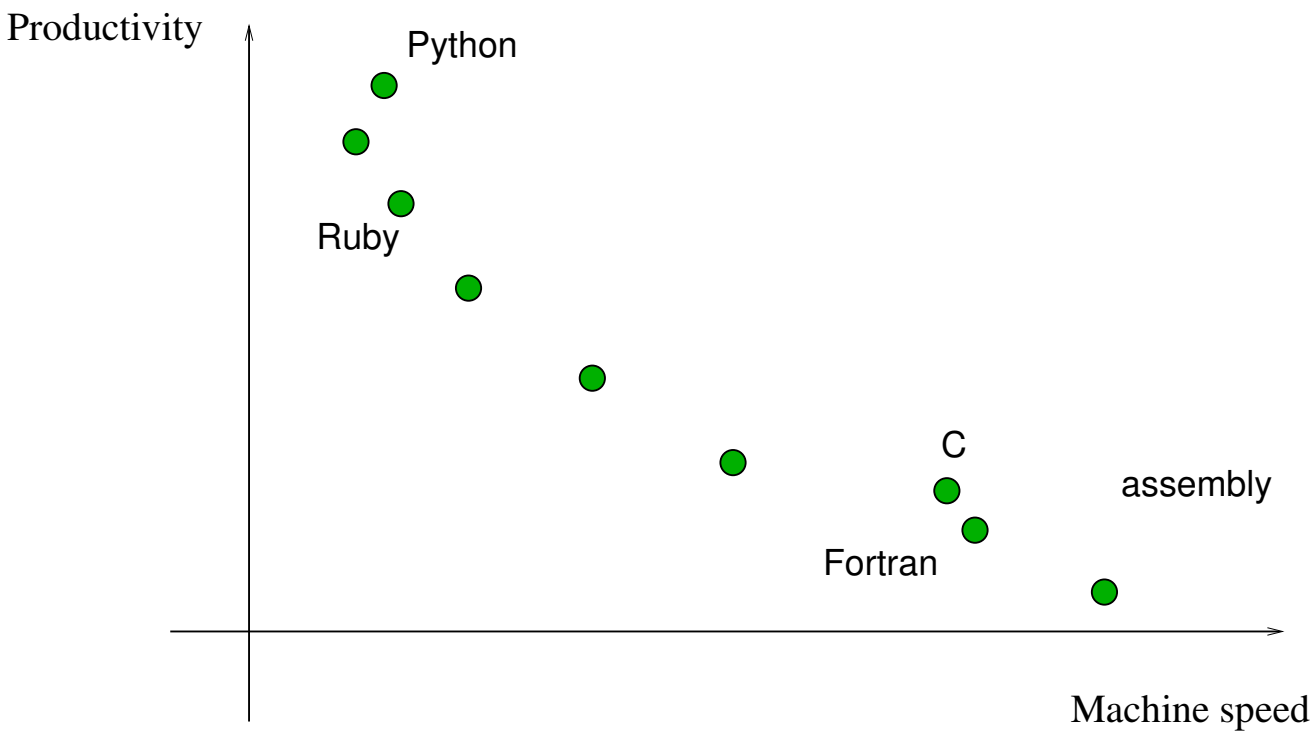
Example. Reading from a file in Python

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

Jane Street on readability:

> *There is no faster way for a trading firm to destroy itself than to deploy a piece of trading software that makes a bad decision over and over in a tight loop.*
>
> *Part of Jane Street's reaction to these technological risks was to put a very strong focus on building software that was easily understood—software that was readable.*
>
> – Yaron Minsky, Jane Street

Trade-Offs

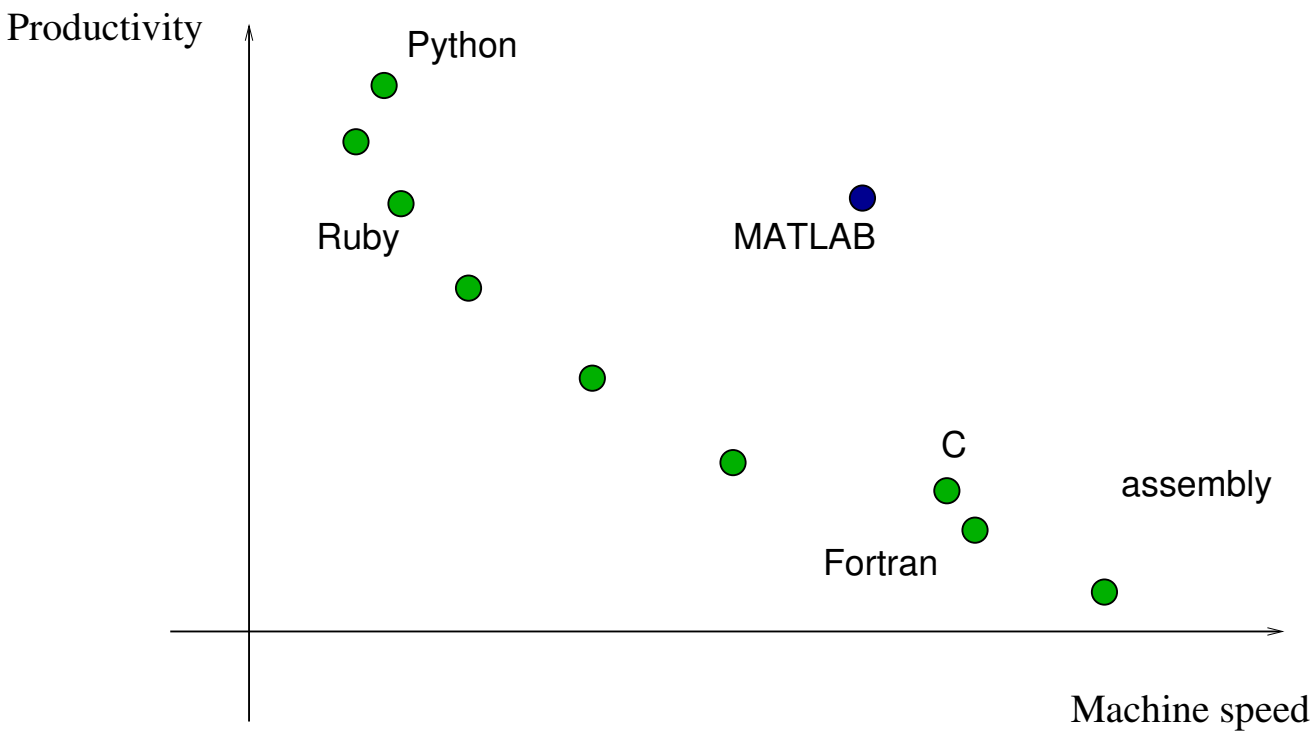

Productivity
Python
Ruby
C
assembly
Fortran
Machine speed

But what about scientific computing?

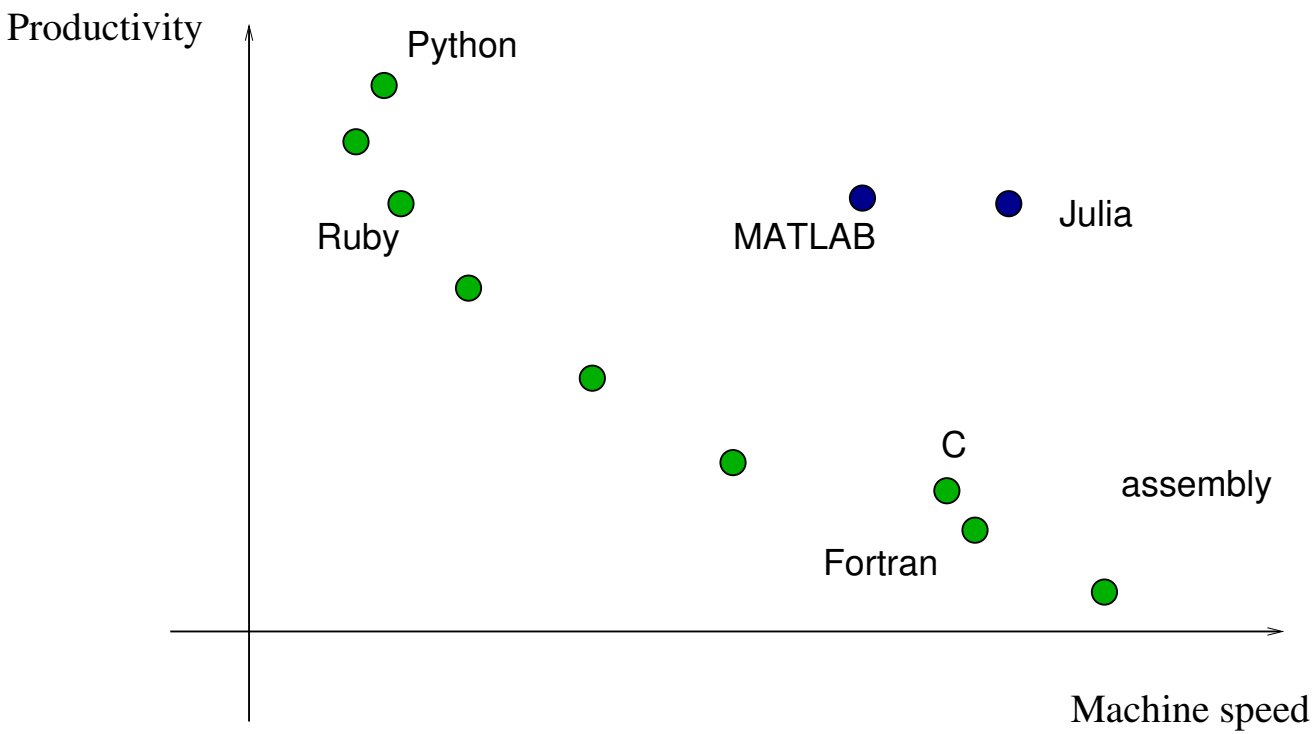
Requirements

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

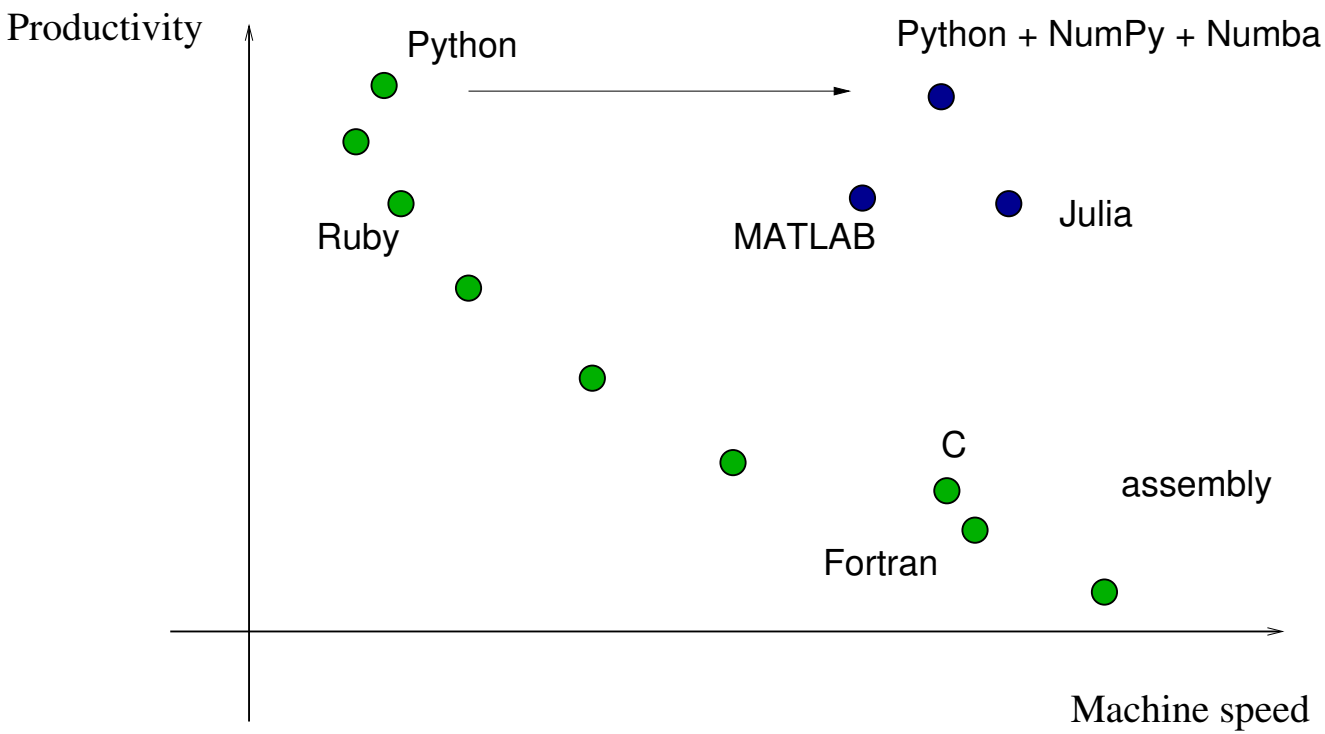
Trade-Offs


Productivity
Python
Ruby
MATLAB
C
assembly
Fortran
Machine speed

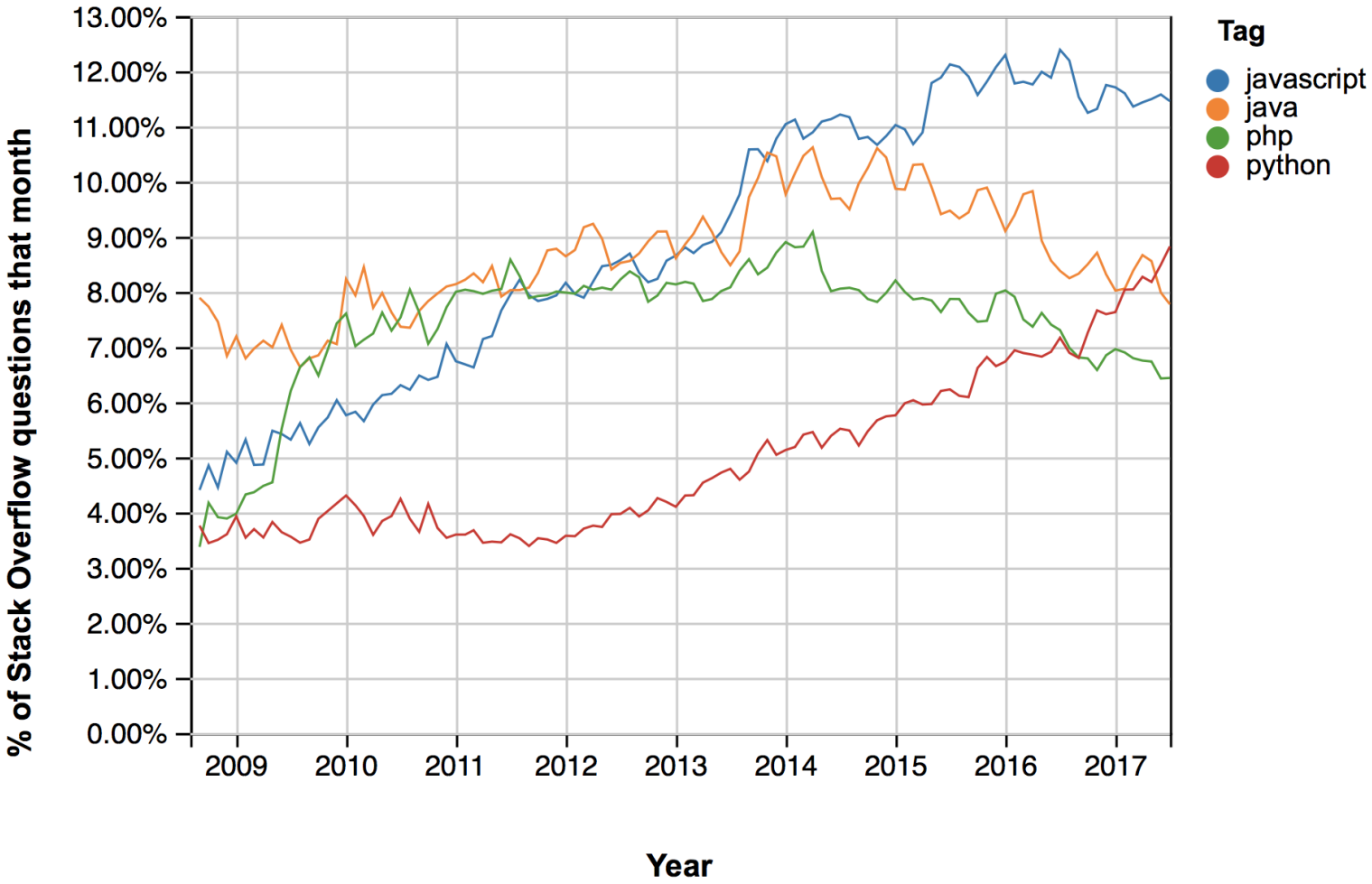
Trade-Offs


Productivity
Python
Ruby
MATLAB
Julia
C
assembly
Fortran
Machine speed

Trade-Offs


Productivity
Python
Python + NumPy + Numba
Ruby
MATLAB
Julia
C
assembly
Fortran
Machine speed

Python vs other popular high level languages



Jupyter notebooks

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

Can be opened through `Anaconda navigator`

Or via a terminal:

Step 1: Open a terminal

- on Windows, use `Anaconda Command Prompt`

Step 2: type `jupyter notebook`

Goals of QuantEcon

QuantEcon wants to help make economists more productive — We do this by developing and documenting open source computational tools for economics, econometrics, and decision making.

Economics Related Projects

- QuantEcon
- Statsmodels
- pyblp
- ...