

Perspective: The State of Julia for Scientific Machine Learning



Edward Berman, Jacob Ginesin
Northeastern University

Tools and Programming Languages

- Python sucks.

Tools and Programming Languages

- Python's adoption is in many ways unnatural:



Tools and Programming Languages

- Python's adoption is in many ways unnatural:
 - Scripting Language



Tools and Programming Languages

- Python's adoption is in many ways unnatural:
 - Scripting Language
 - Slow



Tools and Programming Languages

- Python's adoption is in many ways unnatural:
 - Scripting Language
 - Slow
 - Challenging to maintain



Tools and Programming Languages

- Python's adoption is in many ways unnatural:
 - Scripting Language
 - Slow
 - Challenging to maintain
 - Poor package management



Tools and Programming Languages

- To Reiterate: Python sucks.

Tools and Programming Languages

- To Reiterate: Python sucks.
- **So, why do we use it?**

Tools and Programming Languages

- To Reiterate: Python sucks.
- **So, why do we use it?**
- Python is still uber popular

Tools and Programming Languages

- To Reiterate: Python sucks.
- **So, why do we use it?**
- Python is still uber popular
- If you read a paper here, Python was probably used for the research

Tools and Programming Languages

- A lot of this boils down to the

“TWO LANGUAGE PROBLEM”

Tools and Programming Languages



- The two language problem states that a programming language can not be fast and high-level
- Python is high-level, and often wraps around C/C++ for anything that needs to be performant (e.g. Torch)

Tools and Programming Languages

- In Response, Julia was invented to defy the following laws of nature ([Bezanson et al.](#))
 - High-level dynamic programs have to be slow
 - One must prototype in one language and deploy in another
 - Some things should be left for the experts

Tools and Programming Languages

- Julia directly answers the two language problem
 - Fast and High level

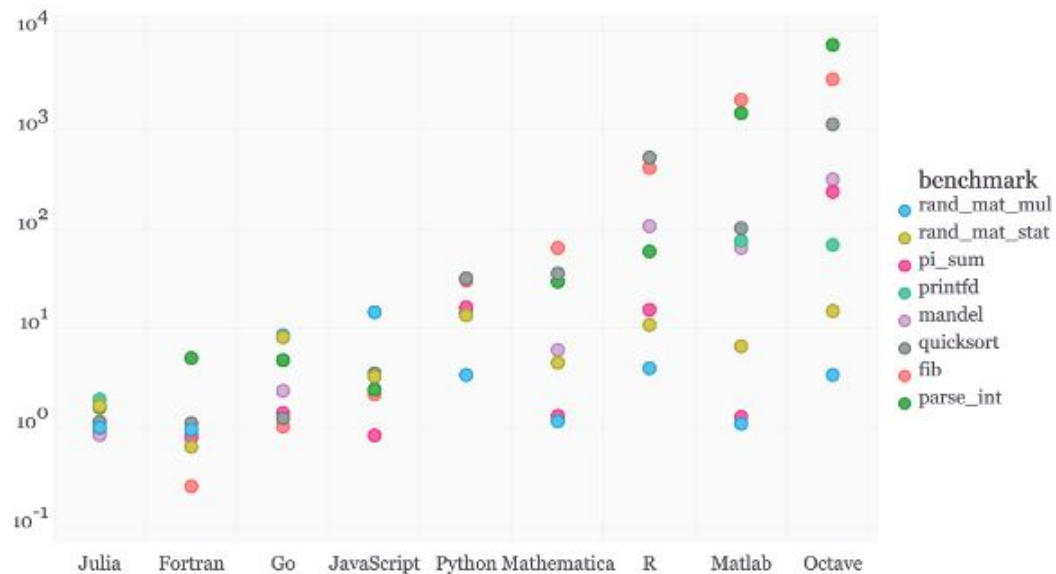


Fig. 5 Performance comparison of various languages performing simple microbenchmarks. Benchmark execution time relative to C. (Smaller is better; C performance = 1.0.)

(Bezanson et al.)

Tools and Programming Languages

- Julia has amazing infrastructure for Scientific Machine Learning
 - SciML, JuMP, Zygote, JuliaDiff, Turing, DifferentialEquations, etc...
 - Great for constrained optimization



Tools and Programming Languages

- Julia also has great package management (Pkg)



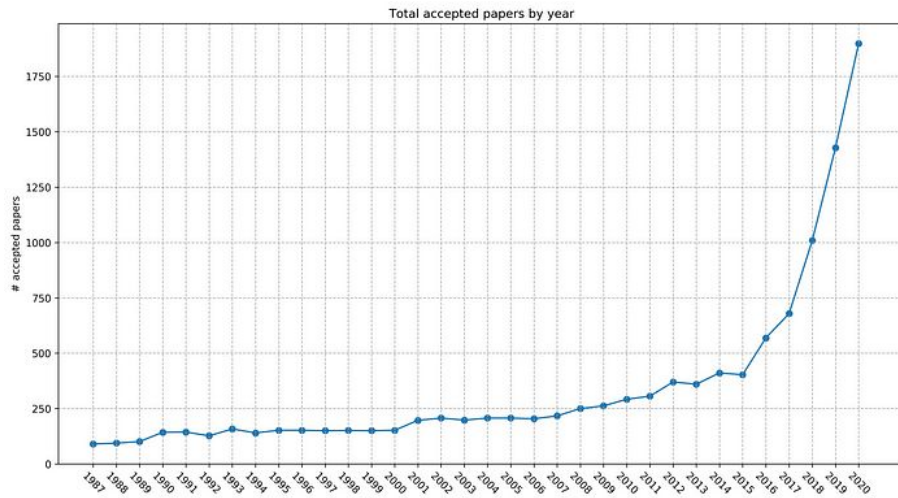
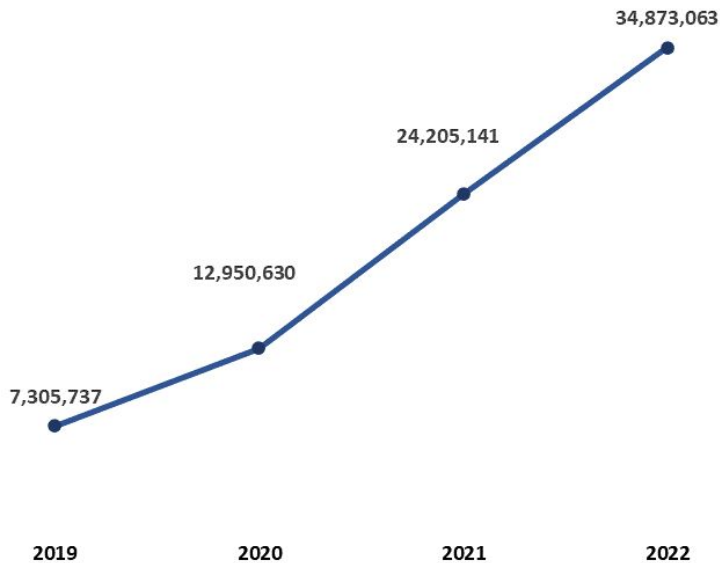
PKG. ADD(**julia**)

Tools and Programming Languages

- Julia is... growing?...

Julia Downloads and Accepted NeurIPS papers over time

Cumulative Julia Downloads As Of Jan 1

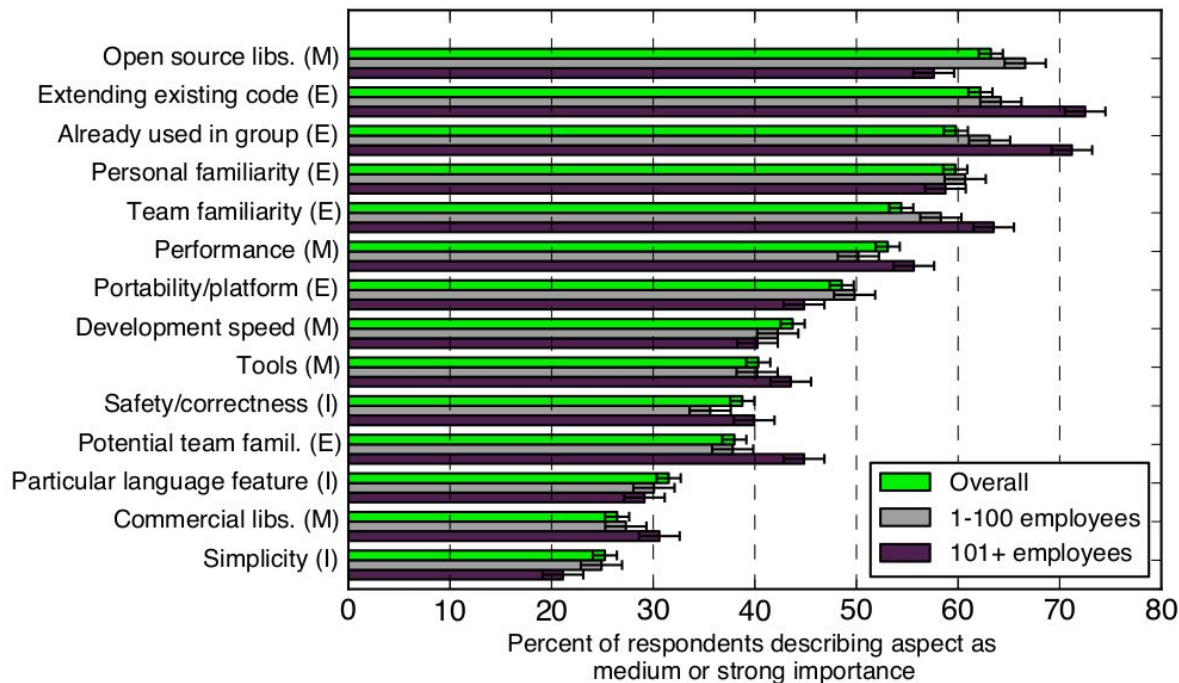


Tools and Programming Languages

- Why is Python still the de-facto language for Scientific Machine Learning Problems? Is the answer really just **momentum**?

Language Adoption

- Clearly momentum plays some role...



[meyerovich et al.]

Language Adoption

- The momentum idea is **incomplete**
- Programming languages like Rust have emerged, despite other languages having similar utility



Language Adoption

- So what gives?

Language Adoption

- So what gives?
- Julia has an amazing ecosystem with stellar performance

Language Adoption

- So what gives?
- Julia has an amazing ecosystem with stellar performance
- Julia has exquisite support for the sciences in particular

Language Adoption

- So what gives?
- Julia has an amazing ecosystem with stellar performance
- Julia has exquisite support for the sciences in particular
- PyTorch even has a Julia dependency! (PySR -> SymbolicRegressions.jl)

Language Adoption

- So what gives?
- Julia has an amazing ecosystem with stellar performance
- Julia has exquisite support for the sciences in particular
- PyTorch even has a Julia dependency! (PySR -> SymbolicRegressions.jl)
- **Why aren't people using Julia?**

Language Adoption

- Julia lacks the following to truly emerge:
 - Industry Support

Language Adoption

- Julia lacks the following to truly emerge:
 - Industry Support
 - Strong testing infrastructure and software engineering features

Language Adoption

- Julia lacks the following to truly emerge:
 - Industry Support
 - Strong testing infrastructure and software engineering features
 - Interoperability (especially with C)

Language Adoption

- Julia lacks the following to truly emerge:
 - Industry Support
 - Strong testing infrastructure and software engineering features
 - Interoperability (especially with C)
 - A good debugger

Language Adoption

- Industry Support (it's the economy stupid!)
 - Julia doesn't have nothing ([case studies](#))

Language Adoption

- Industry Support (it's the economy stupid!)
 - Julia doesn't have nothing (case studies)
 - But it doesn't compare to the competition

Language Adoption

- Industry Support (it's the economy stupid!)
 - Julia doesn't have nothing ([case studies](#))
 - But it doesn't compare to the competition



Language Adoption

- Users find this to be an issue

Language Adoption

- Users find this to be an issue
- Julia Hub Survey Data [\[JuliaCon 2024\]](#) finds:
 - 64% respondents said there are not enough Julia users in field or industry

Language Adoption

- Users find this to be an issue
- Julia Hub Survey Data [\[JuliaCon 2024\]](#) finds:
 - 64% respondents said there are not enough Julia users in field or industry
 - 71% use Julia for research, but only 16% for business critical tasks

Language Adoption

- And why doesn't industry use Julia?

Language Adoption

- And why doesn't industry use Julia?
- Why is it an academic thing?

Language Adoption

- One Potential Answer is Testing and Engineering Features
- Businesses want their software to be reliable!



Language Adoption

- Interoperability is also a big issue

Language Adoption

- Interoperability is also a big issue
- Most users will be coming from Python or C/C++

Language Adoption

- Interoperability is also a big issue
- Most users will be coming from Python or C/C++
- Backwards compatibility is paramount

Language Adoption

- The 2024 JuliaCon survey also identified the debugger and precompile times as problematic

Call to Action

- Try Julia!!!


Call to Action

- Try Julia!!!
- We want the best for Julia and the Julia Community

Call to Action

- Try Julia!!!
- We want the best for Julia and the Julia Community
- We think these issues we have raised are currently not being addressed

Call to Action

Try  !!!

Fin!

- UG applying to Phd programs

<https://ebrmn.space/>

<https://jakegines.in/>