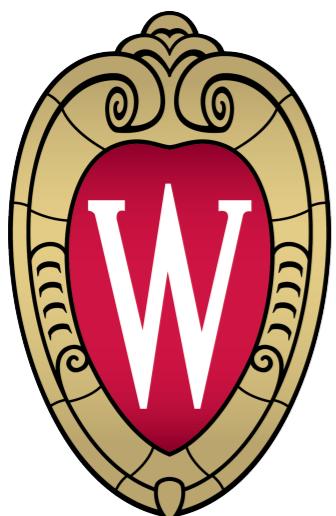


Why julia?



Claudia Solís-Lemus, PhD
University of Wisconsin-Madison
Wisconsin Institute for Discovery
Department of Plant Pathology



<https://solislemuslab.github.io/> @solislemuslab crsl4

Why julia was created

“In short, because we are greedy.

We want a language that's

- open source
- with the speed of C
- obvious, familiar mathematical notation like Matlab
- as usable for general programming as Python
- as easy for statistics as R
- as natural for string processing as Perl
- as powerful for linear algebra as Matlab
- as good at gluing programs together as the shell
- dirt simple to learn, yet keeps the most serious hackers happy”

— Jeff Bezanson, Stefan Karpinski, Viral B. Shah, Alan Edelman

Julia makes headlines

Programming languages: Julia touts its speed edge over Python and R

Benchmarks suggest programming language Julia may be the best choice for big-data analysis using CSV format files.



By [Liam Tung](#) | June 22, 2020 -- 13:38 GMT (06:38 PDT) | Topic:
[Enterprise Software](#)

<https://www.zdnet.com/article/programming-languages-julia-touts-its-speed-edge-over-python-and-r/>



<https://solislemuslab.github.io/>



@solislemuslab



crsl4

Julia makes headlines



Parallel Supercomputing for Astronomy

Researchers use Julia on a NERSC supercomputer (650,000 cores) to speed astronomical image analysis 1,000x, catalog 188 million astronomical objects in 15 minutes and achieve peak performance of 1.5 petaflops per second

Julia makes headlines

Programming language Julia version 1.5 is out: Lots of new features, better performance

Julia programming language gets a new default package manager and easier bug reporting.



By [Liam Tung](#) | August 3, 2020 -- 12:41 GMT (05:41 PDT) | Topic:
[Enterprise Software](#)

<https://www.zdnet.com/article/programming-language-julia-version-1-5-is-out-lots-of-new-features-better-performance/>



<https://solislemuslab.github.io/>



@solislemuslab



crsl4

julia comparison

	Fortran gcc 5.1.1	Julia 0.4.0	Python 3.4.3	R 3.2.2	Matlab R2015b	Octave 4.0.0	Mathematica 10.2.0
fib	0.70	2.11	77.76	533.52	26.89	9324.35	118.53
parse_int	5.05	1.45	17.02	45.73	802.52	9581.44	15.02
quicksort	1.31	1.15	32.89	264.54	4.92	1866.01	43.23
mandel	0.81	0.79	15.32	53.16	7.58	451.81	5.13
pi_sum	1.00	1.00	21.99	9.56	1.00	299.31	1.69
rand_mat_stat	1.45	1.66	17.93	14.56	14.52	30.93	5.95
rand_mat_mul	3.48	1.02	1.14	1.57	1.12	1.12	1.30

Figure: benchmark times relative to C (smaller is better, C performance = 1.0).



<https://solislemuslab.github.io/>



@solislemuslab



crsl4

julia comparison

Test	R 3.4.1	Julia 0.6.0	Speedup
Matrix creation, trans., deform. (2500 x 2500)	0.17	0.25	0.67
Power of matrix (2500 x 2500, $A.^{1000}$)	0.55	0.25	2.18
Quick sort ($n = 7 \times 10^6$)	0.68	0.59	1.15
Cross product (2800 x 2800, $A^T A$)	6.15	0.20	31.15
LS solution ($n = p = 2000$)	12.83	0.15	88.34
FFT ($n = 2,400,000$)	0.32	0.12	2.59
Eigen-values (600 x 600)	0.65	0.49	1.33
Determinant (2500 x 2500)	2.47	0.12	20.00
Cholesky (3000 x 3000)	2.87	0.14	20.00
Matrix inverse (1600 x 1600)	5.04	0.17	29.87
Fibonacci (vector calculation)	0.22	0.19	1.17
Hilbert (matrix calculation)	0.23	0.06	3.69
GCD (recursion)	0.39	0.08	4.64
Toeplitz matrix (loops)	0.04	0.00	49.69
Escoufiers (mixed)	0.33	0.15	2.14

Machine specs: Intel i7 (Skylake) @ 2.9GHz (4 physical cores, 8 threads), 16G RAM, Mac OS Sierra 10.12.6.

https://github.com/Hua-Zhou/SGSC2017Colorado/blob/master/notebooks/13_Julia_Overview/13_Julia_Overview.ipynb



<https://solislemuslab.github.io/>

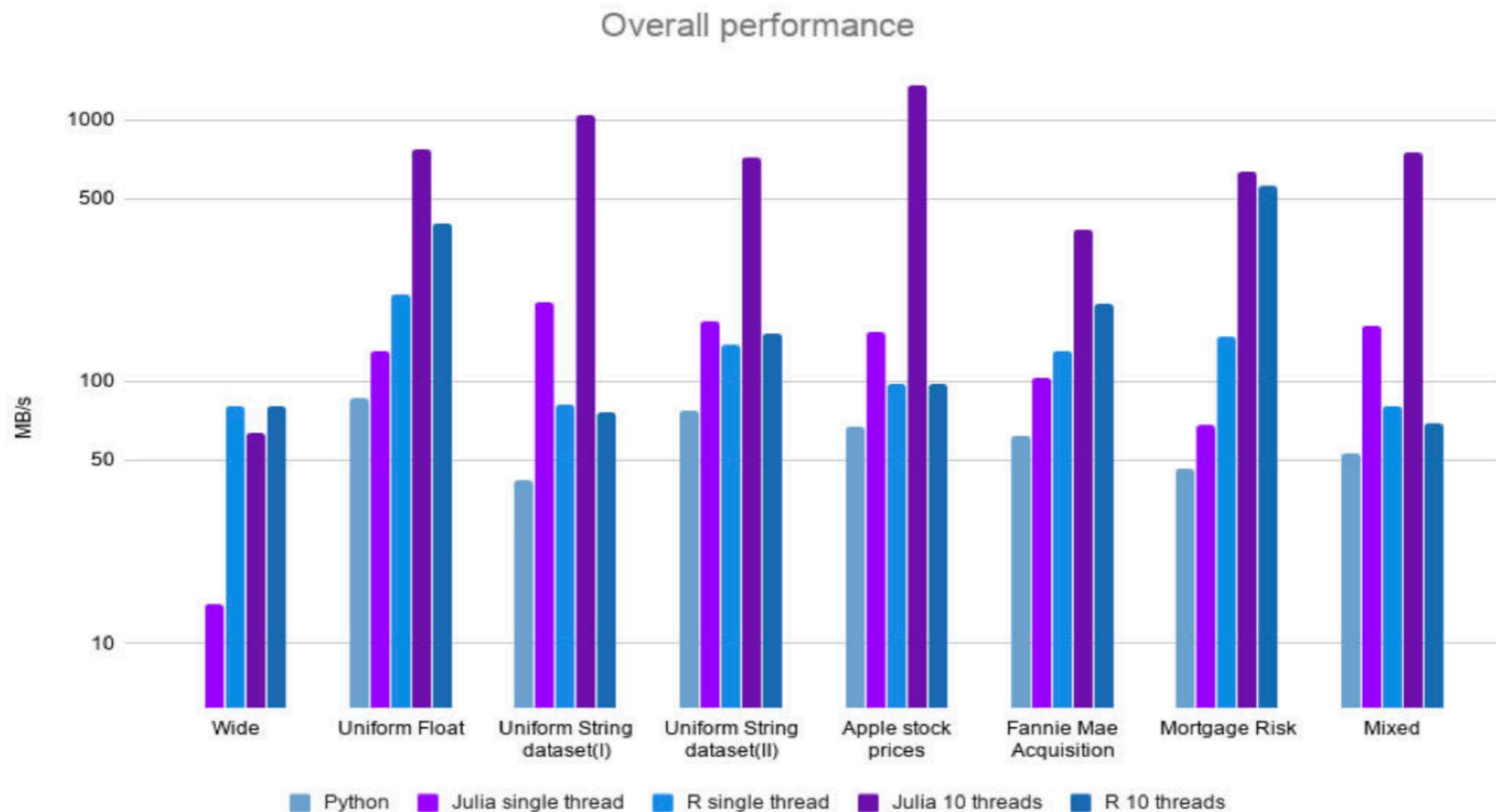


@solislemuslab



crsl4

julia comparison



Julia Computing says, across all eight datasets, Julia's CSV.jl is always faster than Pandas, and with multi-threading it is competitive with R's data.table.

Image: Julia Computing

Why did I choose Julia?

Short answer: Because of Doug Bates

As some of you may know, I have had a (rather late) mid-life crisis and run off with another language called Julia.

— Doug Bates



Comparative timings

- In *R* generating a bivariate sample of size 10,000 with a thinning of 500 takes about 97 sec. on this laptop

```
> system.time(Rgibbs(10000,500))  
  user  system elapsed  
R 96.740    0.004   97.027
```

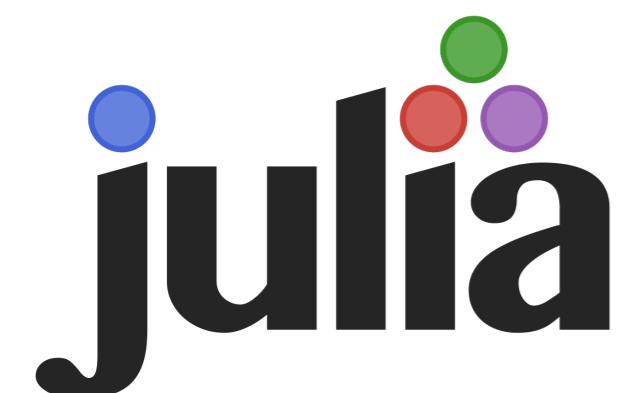
- The corresponding *Julia* function runs in less than 1 second.

```
julia> jgibbs(10000,500); # warm-up  
julia> @elapsed jgibbs(10000,500)  
0.867915085 julia
```

- In *Julia* the first call to a method invokes the JIT so we time the second call.



```
Rgibbs <- function(N,thin) {  
  mat <- matrix(0,nrow=N,ncol=2)  
  x <- y <- 0  
  for (i in 1:N) {  
    for (j in 1:thin) {  
      x <- rgamma(1,3,y*y + 4) # 3rd arg is rate  
      y <- rnorm(1,1/(x + 1),1/sqrt(2*(x + 1)))  
    }  
    mat[i,] <- c(x,y)  
  }  
  mat  
}
```



```
using Distributions  
function jgibbs(N::Integer, thin::Integer)  
  mat = Array(Float64,(N,2))  
  x = y = 0.  
  for i in 1:N  
    for j in 1:thin  
      x = rand(Gamma(3.,1./(y*y+4.))) #shape/scale  
      y = rand(Normal(1./(x+1.),1./sqrt(2.(x+1.))))  
    end  
    mat[i,1] = x; mat[i,2] = y  
  end  
  mat  
end
```



<https://solislemuslab.github.io/>



@solislemuslab



crsl4



PhyloNetworks: analysis for phylogenetic networks in **Julia**

Maximum pseudolikelihood estimation of species network: SNaQ

[build](#) passing [docs](#) [stable](#) [docs](#) [latest](#)

SNaQ implements the statistical inference method in Solís-Lemus and Ané (2016, *PLoS Genetics*). The procedure involves a numerical optimization of branch lengths and inheritance probabilities and a heuristic search in the space of phylogenetic networks.



Why did I stick to Julia?

- I was used to having
 - C++ code
 - R code
 - perl code
 - shell code
- Now I do everything in Julia
- All Julia packages are on Github: very easy to understand code, and to contribute
- Access shell and R from Julia



<http://solislemuslab.github.io/>



@solislemuslab



crsl4

julia vs R

- R is great:
 - open source
 - 3000+ packages in CRAN
 - many resources to learn
- R is great, **but**:
 - for/while are slow
 - not easy to do parallelization, memory mapping
 - R functions do not modify arguments: terrible if you have big objects. Often the key to speeding up an algorithm is cutting down on the copies being created.
 - The real work is done in underlying C, and it is not so easy to track it down

julia resources to get started

- For motivation: <https://juliacomputing.com/case-studies/>
- For general knowledge:
 - <https://learnxinyminutes.com/docs/julia/>
 - <https://github.com/dmbates/NESS2017>
 - <https://julialang.org/learning/>
 - <https://repsychling.github.io/tutorial.html>
 - <https://github.com/smlp2020-stream4/SMLP2020/>
- Information about packages: <https://juliaobserver.com/>
- For genetics: <https://github.com/Hua-Zhou/SGSC2017Colorado/tree/master/notebooks>



<https://solislemuslab.github.io/>



@solislemuslab



crsl4

Where is julia going?

JuliaHub

Track the pulse of the Julia ecosystem, find the packages you use, and discover new packages.

[Log in](#) to access additional functionality like registering packages.

Cloud computing is coming to JuliaHub. [Get notified](#) when it's ready.

🔥 Popular Packages

Flux
IJulia
Gadfly
Pluto
DifferentialEquations
Gen
JuMP
Knet
Plots

Popular Tags

lightgraphs data-structures interpolation
time-series regression neural-ode robotics
data-science differentialequations economics
deep-learning automatic-differentiation flux
differential-equation siml sde arrays
queryverse optimization math finance
mcmc machine-learning data
dde gpu simulation simulation
scientific-machine-learning dae
plotting ode statistics bioinformatics
bayesian-inference nlp hacktoberfest
graph visualization linear-algebra astronomy
neural-networks physics dynamical-systems time-series-analysis
physics python

JuliaHub
beta



<https://solislemuslab.github.io/>



@solislemuslab



crsl4

How to code in julia ?

(I cannot live without RStudio)



<https://solislemuslab.github.io/>



@solislemuslab



crsl4

How to code in julia ?

Editors and IDEs

Juno



Atom Plugin

VS Code



VS Code Extension

Jupyter



Jupyter kernel

JetBrains



IntelliJ IDEA Plugin

Vim



Vim plugin

Emacs



Emacs plugin

SublimeText



Sublime Text

NotePad++



Notepad++

<https://julialang.org/>



<https://solislemuslab.github.io/>

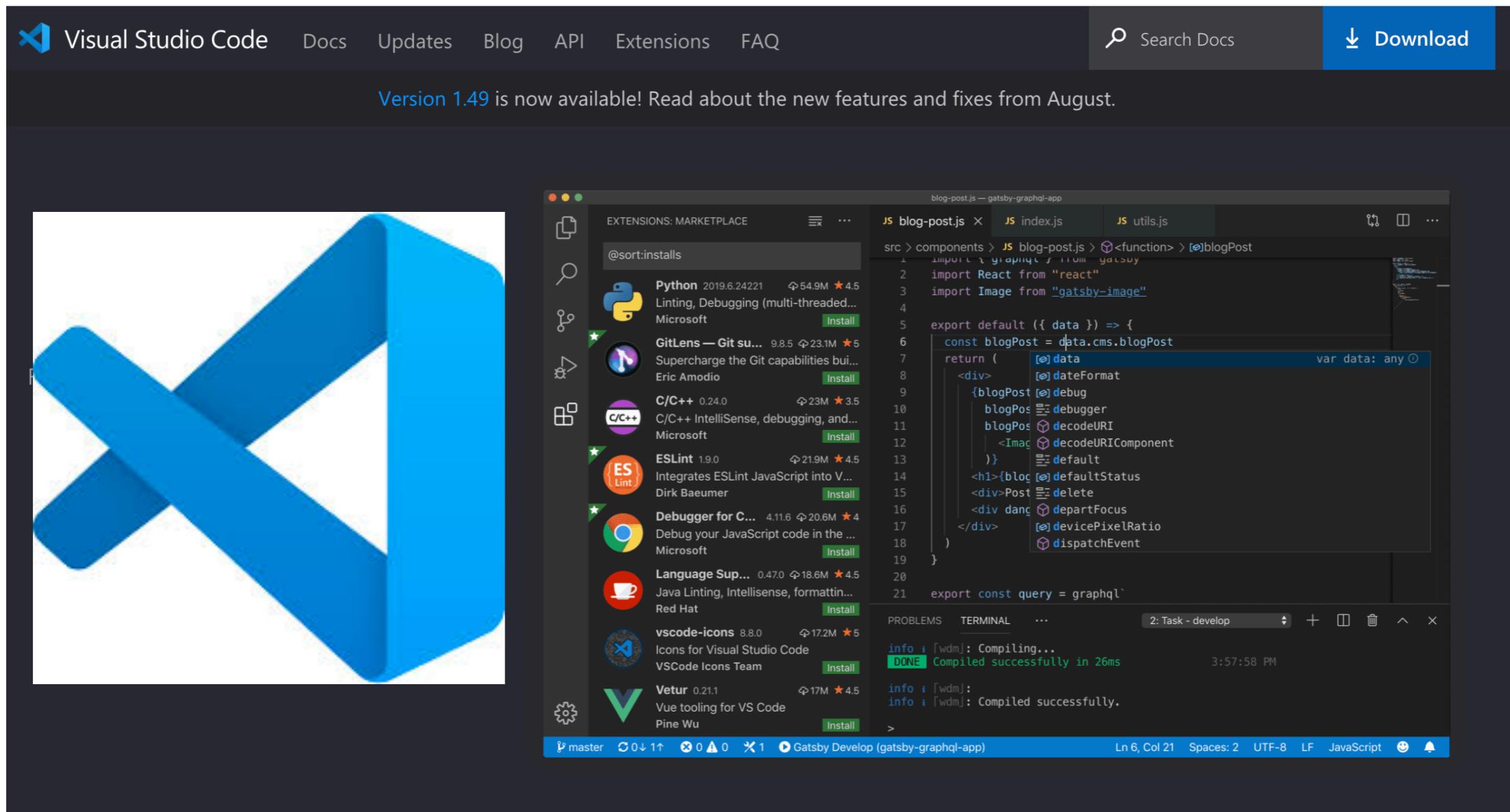


@solislemuslab



crsl4

How to code in julia ?



<https://solislemuslab.github.io/>



@solislemuslab

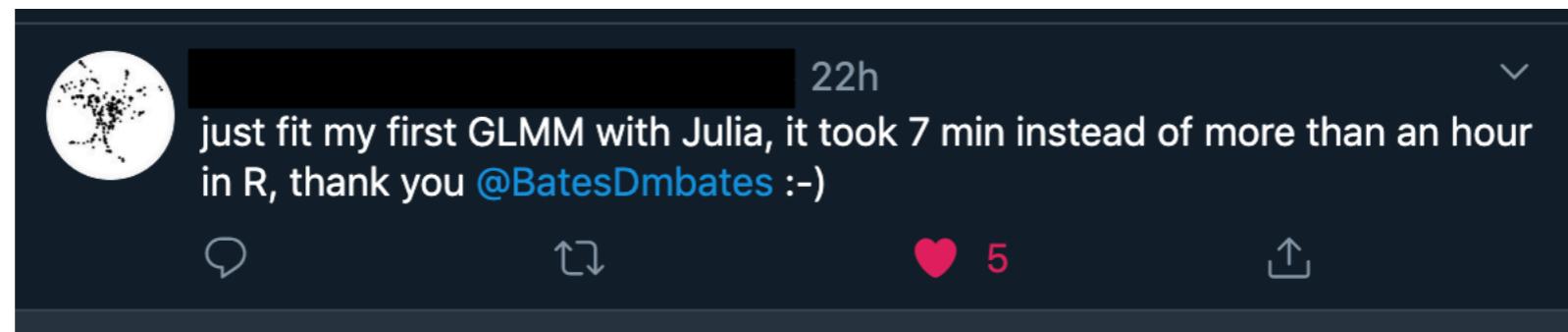


crsl4

Let's get started!



Huge **thanks** to Doug Bates!



<https://solislemuslab.github.io/>



@solislemuslab



crsl4