The state of JuMP

~~Miles Lubin~~ Oscar Dowson

SIAM OP26

What is JuMP?

An algebraic modeling language in Julia

```
using JuMP, Ipopt
function solve_constrained_least_squares_regression(A::Matrix, b::Vector)
    m, n = size(A)
    model = Model(Ipopt.Optimizer)
    @variable(model, -10 <= x[1:n] <= 10)
    @variable(model, residuals[1:m])
    @constraint(model, residuals == A * x - b)
    @constraint(model, sum(x) == 1)
    @objective(model, Min, sum(residuals[i]^2 for i in 1:m))
    optimize!(model)
    assert_is_solved_and_feasible(model)
    return value.(x)
end
x = solve_constrained_least_squares_regression(rand(10, 3), rand(10))
```

Why is JuMP interesting?

Non-technical

- Nice syntax
- Great documentation
- Vibrant community
- Open source
- Easy to install

Technical

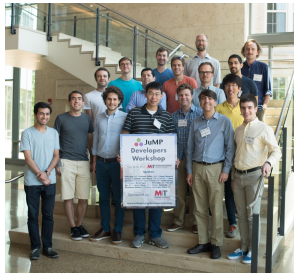
- LP, MIP, NLP, Conic, SDP
- Interact with solvers while they're running
- Low overhead for model generation
- Extensible to new solvers and problem classes

Who is (some of) JuMP?

https://github.com/jump-dev/JuMP.jl/graphs/contributors

JuMP-dev workshops

All talks on YouTube. Go to https://jump.dev



2017, Cambridge

2018, Bordeaux

2019, Santiago

2020

Juliacon 2021 (with JuMP-dev) was online and virtual

28th to 30th of July, 2021
check out all the recorded talks on YouTube

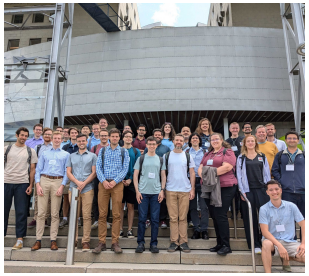
2021, Online

JuliaCon 2022 was online and the recorded talks are available on YouTube

July 27th - 29th - Times in UTC
Tickets were free but registration was required

2022, Online

2023, Cambridge



2024, Montreal

2025, Auckland

2026, Edinburgh

JuMP-dev workshops

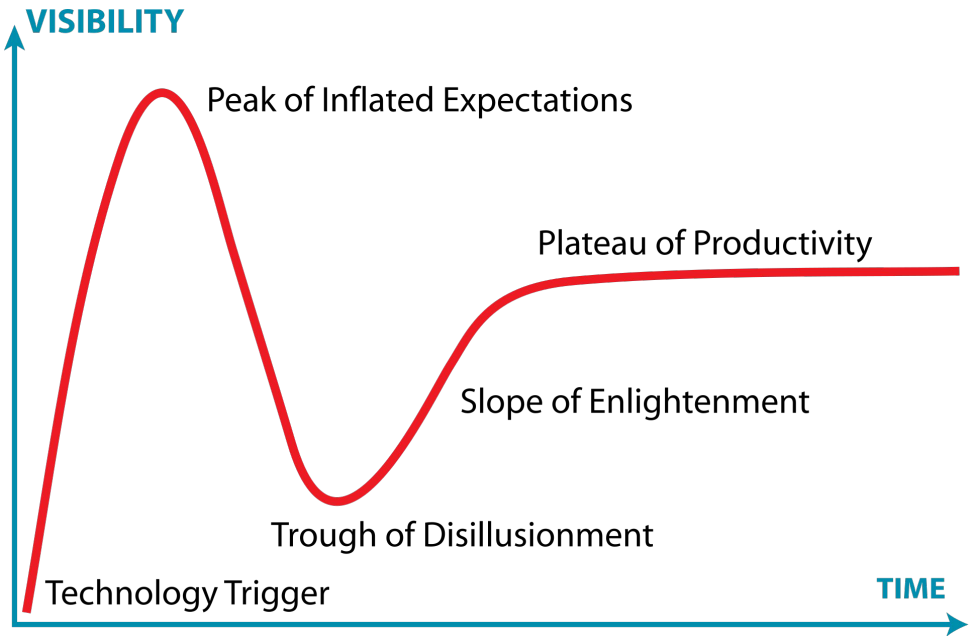
All talks on YouTube. Go to https://jump.dev

The JuMP-dev prize

https://jump.dev/prize/jump-dev-2026

JuMP's stage of development

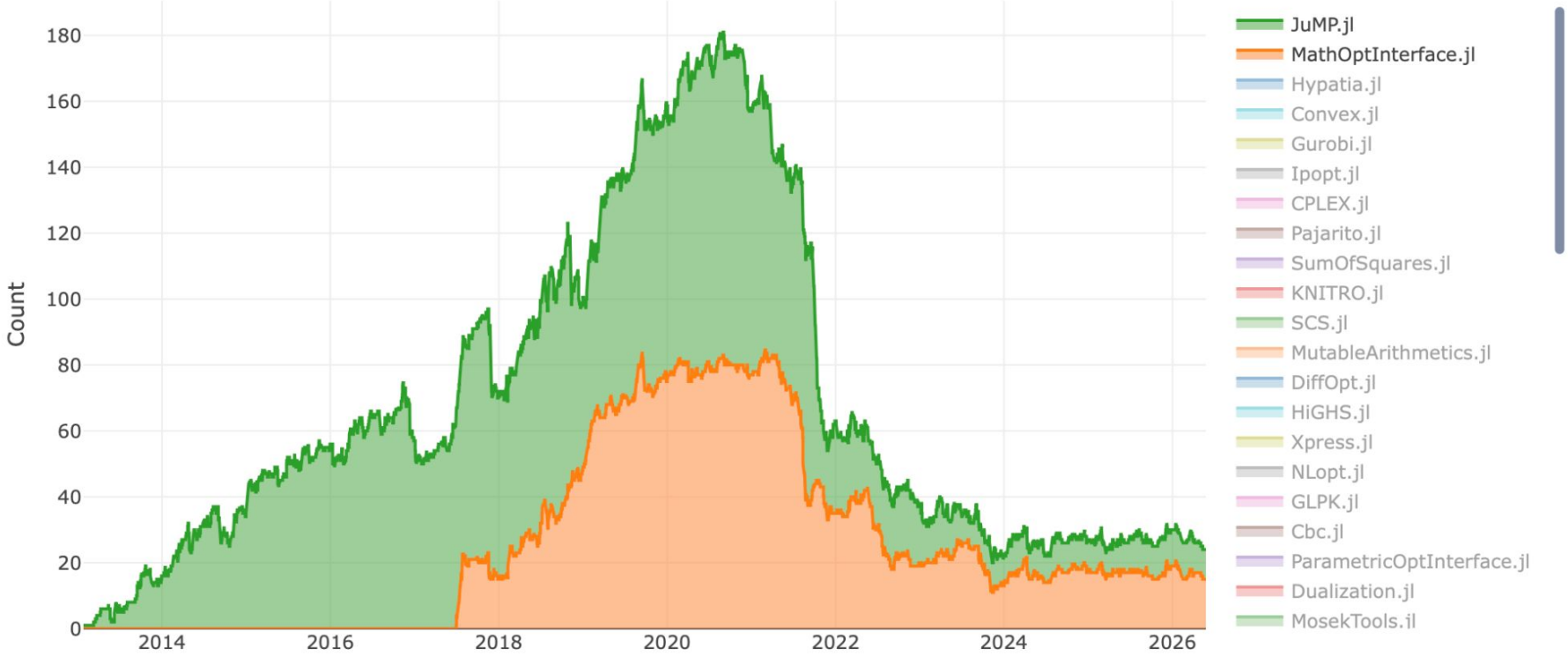
Where are we in the hype cycle?



Source: Jeremykemp (Wikipedia)

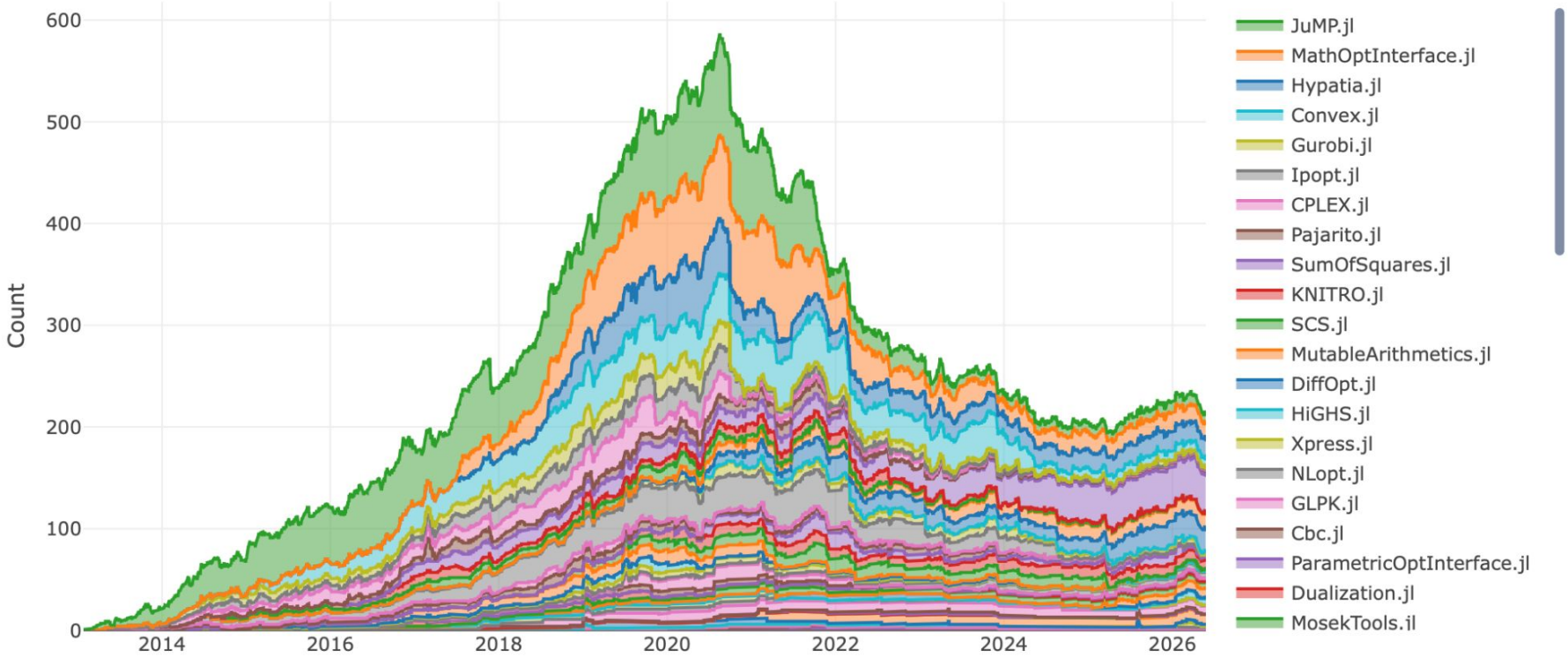
Count of open issues by package

Smaller is better. Open issues are a mixture of bugs and feature requests.



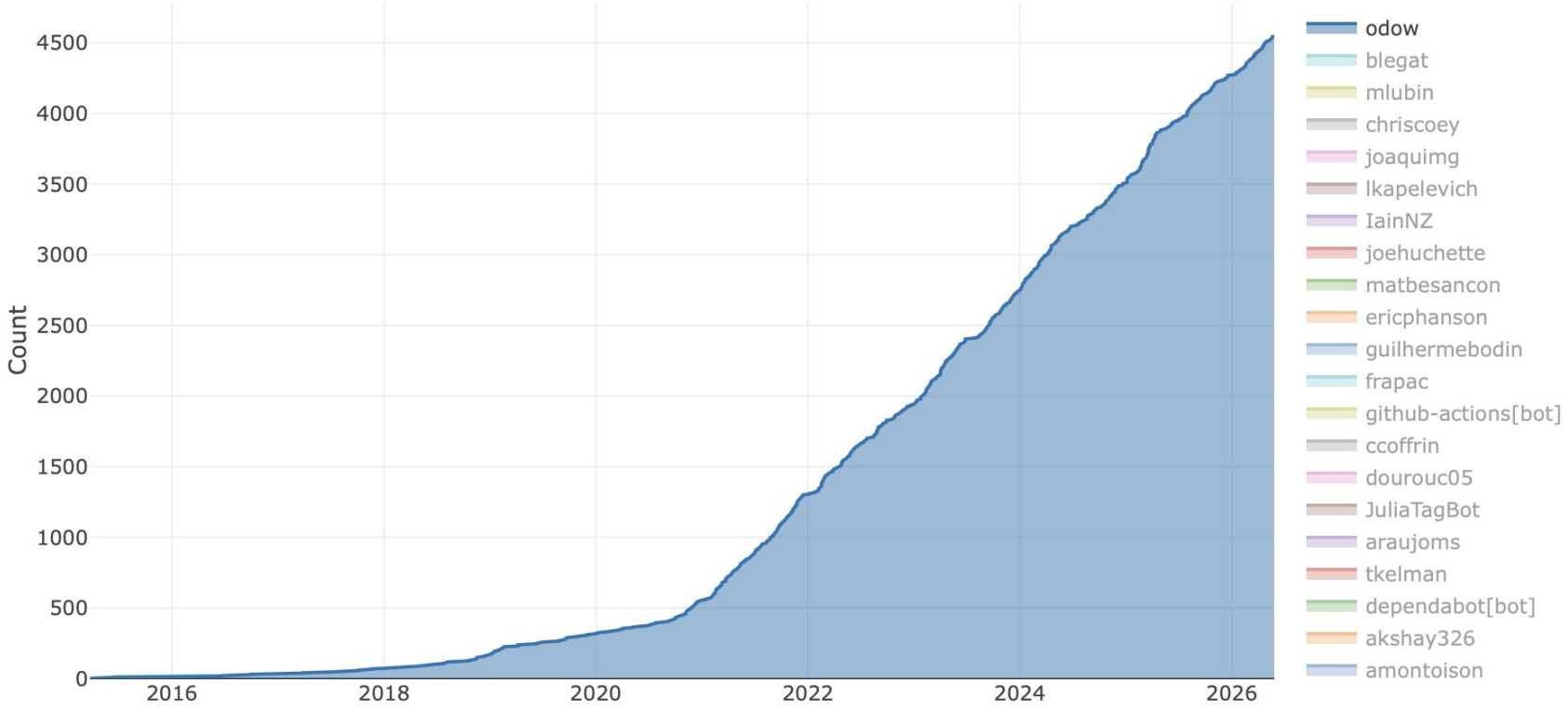
Count of open issues by package

Smaller is better. Open issues are a mixture of bugs and feature requests.



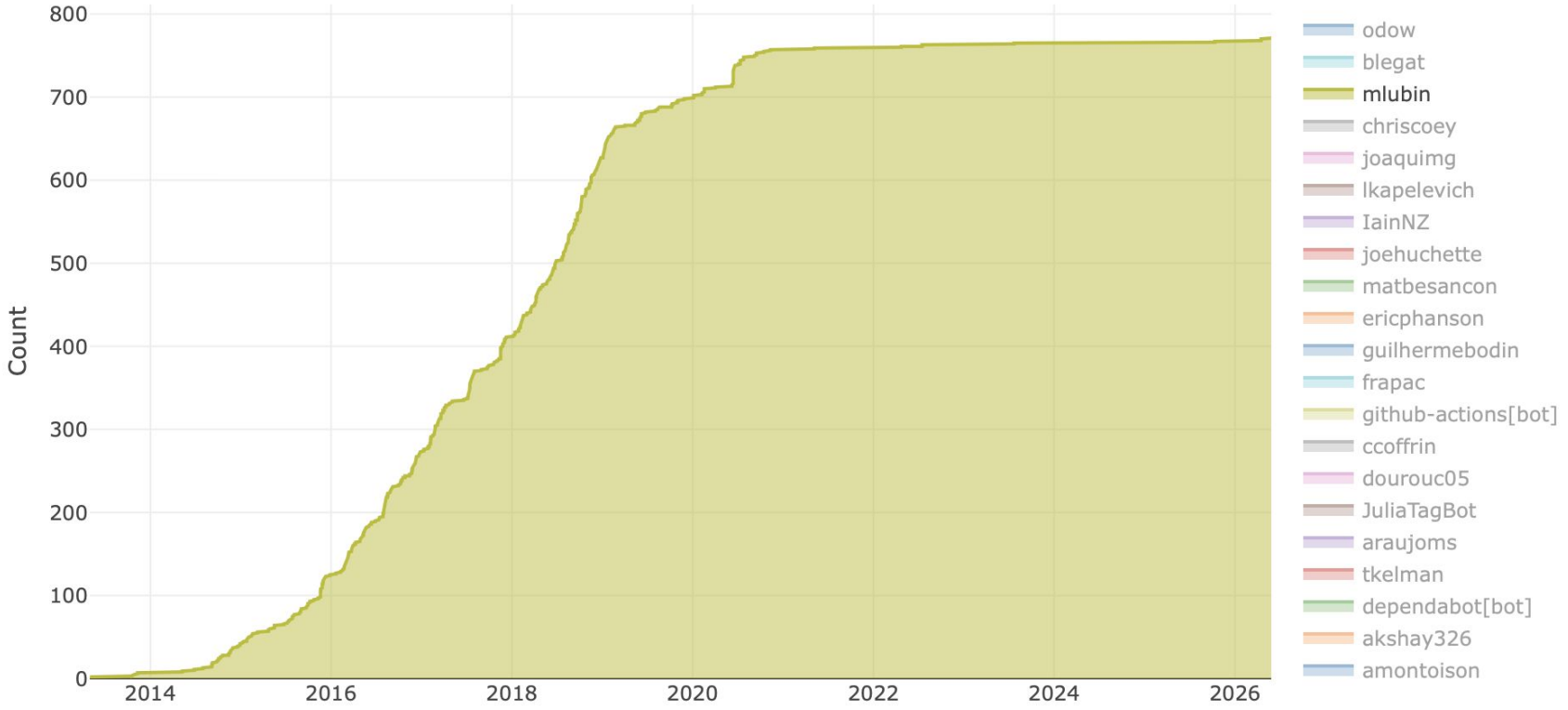
Cumulative count of opened pull requests by user

We are mainly interested in the proportion of work done by each user. The vast majority of users make few contributions.



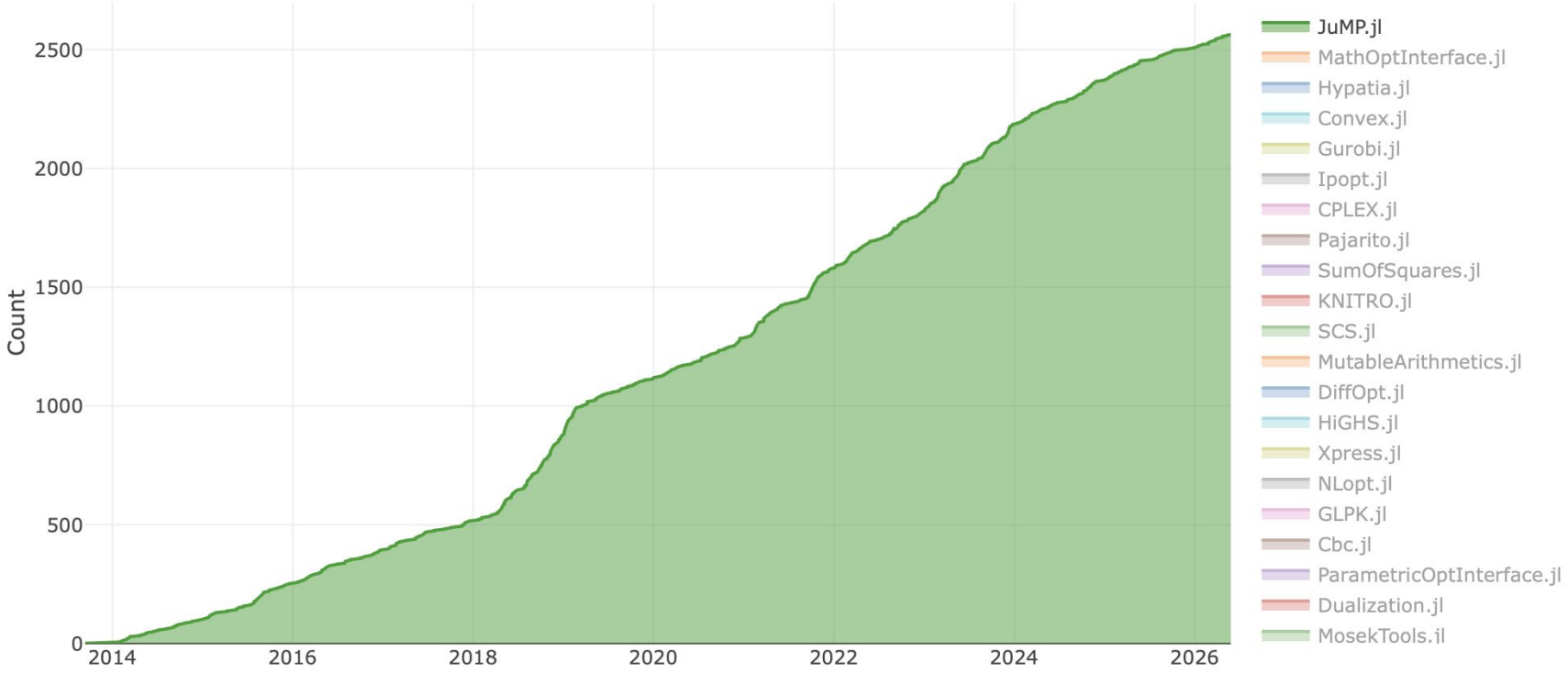
Cumulative count of opened pull requests by user

We are mainly interested in the proportion of work done by each user. The vast majority of users make few contributions.



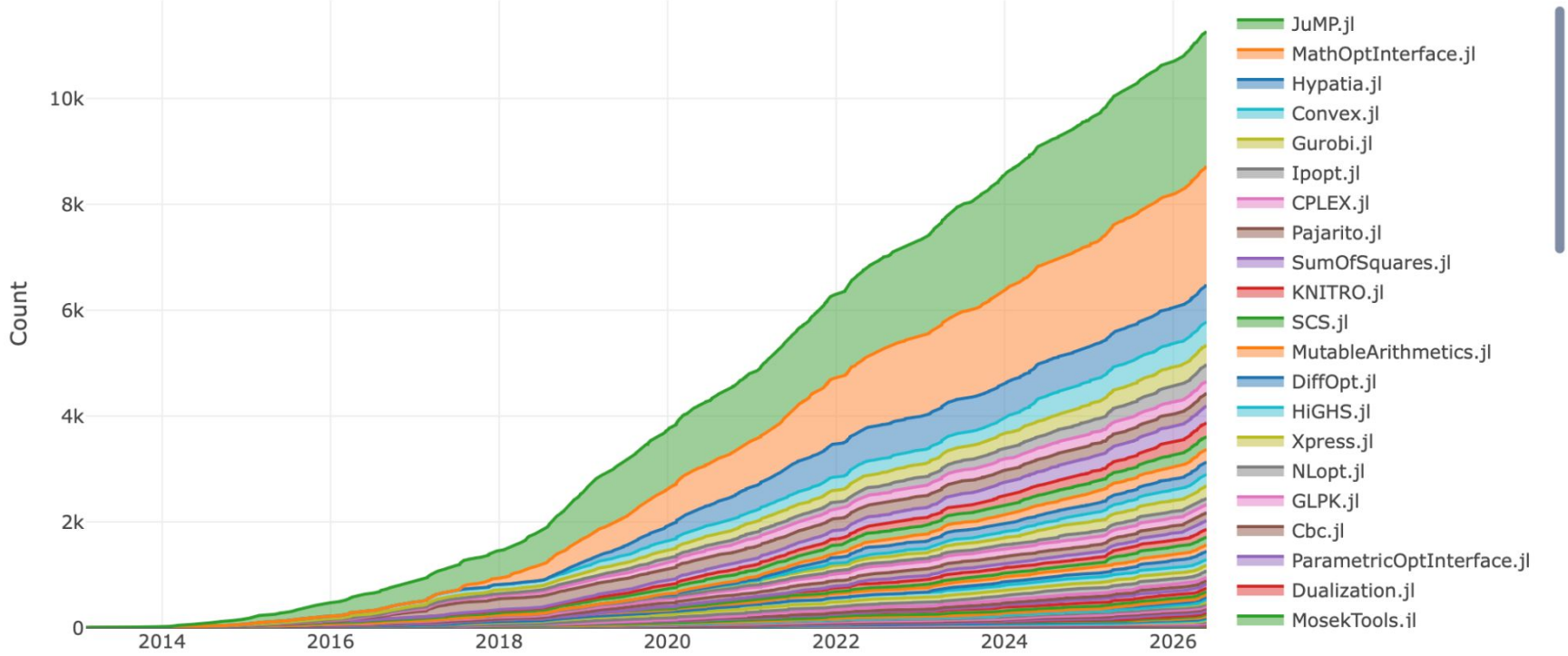
Cumulative count of opened pull requests by package

To be viewed in conjunction with the count of open pull requests. We're more interested in the slope of each line, rather than the absolute numbers.



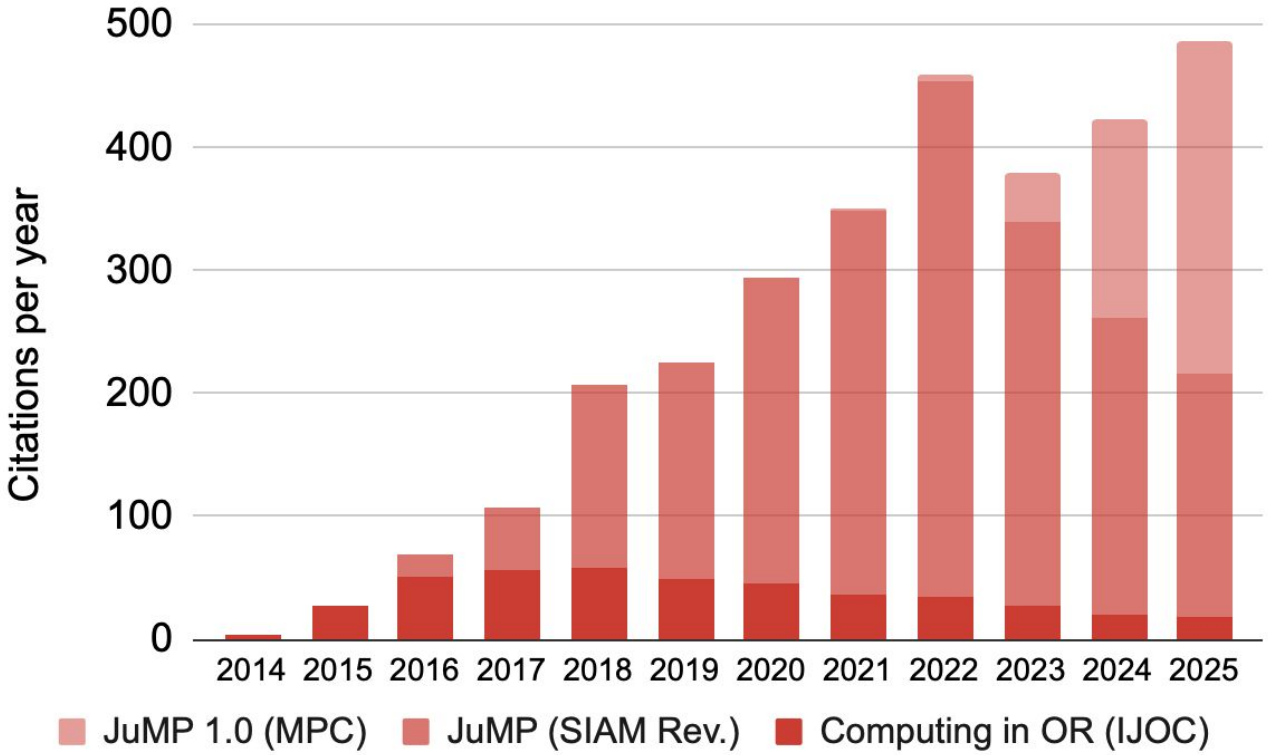
Cumulative count of opened pull requests by package

To be viewed in conjunction with the count of open pull requests. We're more interested in the slope of each line, rather than the absolute numbers.



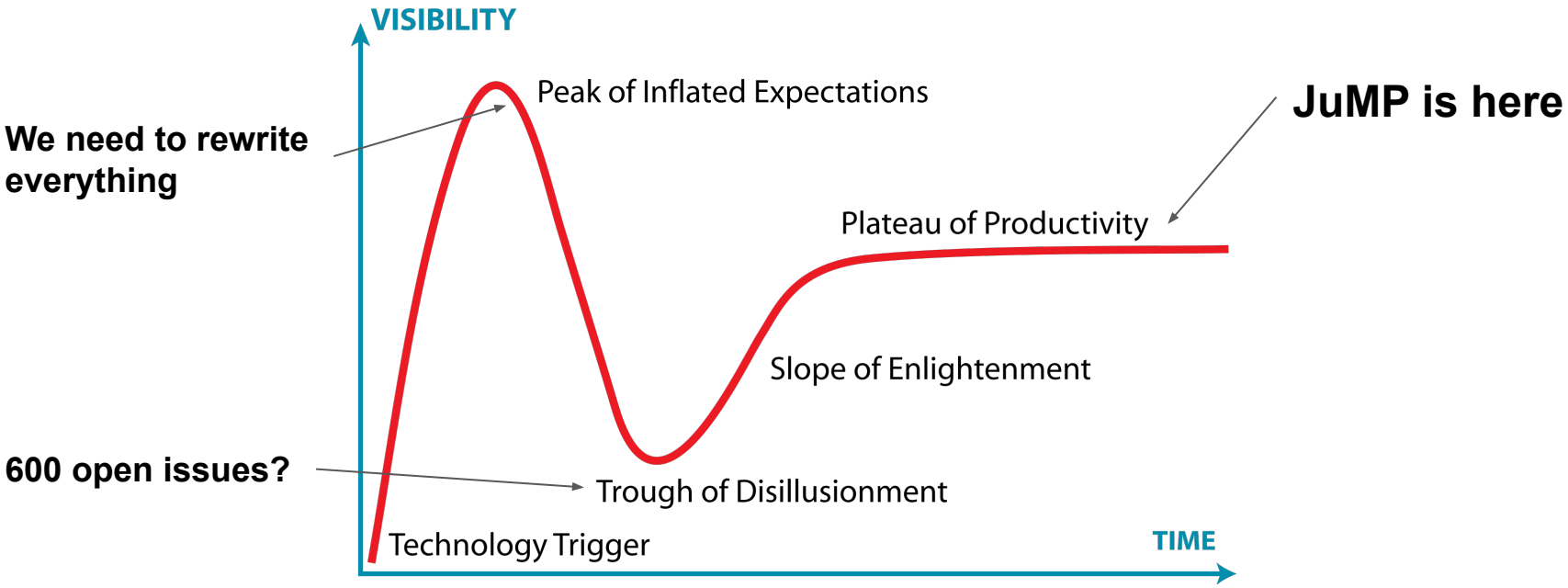
Citations of JuMP papers

Google Scholar citations per year



JuMP's stage of development

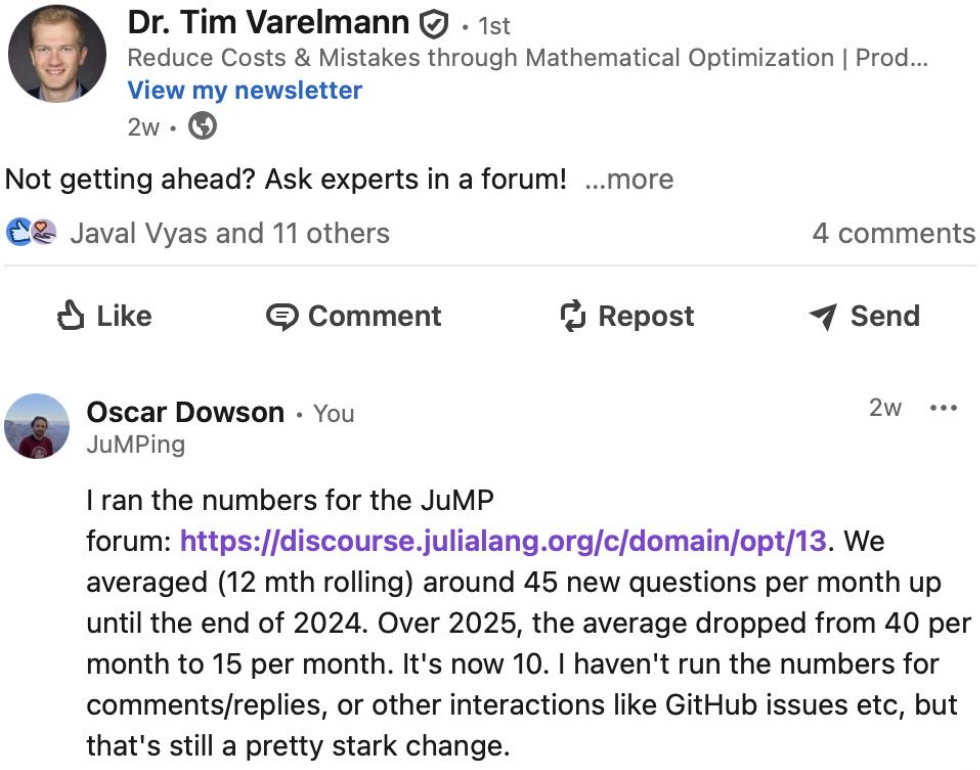
We rode out of the hype cycle



Source: Jeremykemp (Wikipedia)

Support questions are down 80%

I guess this is a good thing?



Dr. Tim Varelmann • 1st
Reduce Costs & Mistakes through Mathematical Optimization | Prod...
View my newsletter
2w •

Not getting ahead? Ask experts in a forum! ...more

Javal Vyas and 11 others — 4 comments

Like Comment Repost Send

Oscar Dowson • You
JuMPing
2w

I ran the numbers for the JuMP forum: https://discourse.julialang.org/c/domain/opt/13. We averaged (12 mth rolling) around 45 new questions per month up until the end of 2024. Over 2025, the average dropped from 40 per month to 15 per month. It's now 10. I haven't run the numbers for comments/replies, or other interactions like GitHub issues etc, but that's still a pretty stark change.

Development is moving to external packages

There are many to check out

jump-dev/MathOptAnalyzer.jl

Check models for common modeling and numerical issues

jump-dev/MathOptComplements.jl

Reformulate complementarity constraints

jump-dev/MathOptIIS.jl

Compute an IIS for an arbitrary JuMP model

jump-dev/MathOptLazy.jl

Solve problems with lazy constraints, iteratively or via callbacks

samuelsonric/MathOptChordalDecomposition.jl

Simplify semidefinite constraints by performing a chordal decomposition.

lanl-ansi/MathOptAI.jl

Embed neural networks into a JuMP model

jump-dev/DiffOpt.jl

Compute derivatives of solutions w.r.t. problem data

jump-dev/ParametricOptinterface.jl

Substitute parameters out of JuMP models

But we're still making improvements to JuMP

A new LP reader, with better error messages

```
julia> using JuMP

julia> read_from_file("model.lp")
ERROR: Error parsing LP file on line 72:
e_group0_t9717p0_1: winS_0 - S_t9717p0 GS_t9717p0 >= 0
                                       ^

Got an identifier with value `GS_t9717p0`. We expected this to be an
inequality like `>=`, `<=`, or `==`.

Stacktrace:
```

GPU solvers

We're seeing lots of activity in this space

Pure Julia versions

- **MadNLP/MadNLP.jl**
- JuliaDecisionFocusedLearning/CoolPDLP.jl
- cvxgrp/CuClarabel

Plus interfaces to other solvers

- MIT-Lu_Lab/CuPDLPx.jl
- jump-dev/cuOpt.jl

Roadmap

Where are we going?

- Batching (See MathOptInterface.jl#2904)
- Scientific Unit support (Pyomo has this)
- ???

Fewer changes isn’t bad. I want JuMP to be “boring”

JuMP's influence

Is broader than jump-dev/JuMP.jl

- JuMP-inspired modeling packages
  - C++/Python: MathOpt (Google OR-Tools)
  - Python: PyOptInterface
  - R: ompr
  - Julia: ExaModels (2023 COIN-OR CUP)
- GPU-enabled solvers
  - MadNLP, CuClarabel
  - google-research/FirstOrderLp.jl inspired cuPDLP/cuPDLPx/HPR-LP
- Exotic cones
  - SCS, Mosek, Hypatia
  - LogDetConeTriangle, ComplexPSDCone, …

Thank you!

Go to jump.dev for more information

Support questions are falling off a cliff

Page views of the JuMP documentation are down 23%