

# JuMP 1.0: Recent improvements to a modelling language for mathematical optimization

Oscar Dowson

ORSNZ 2022

# What is JuMP?

Part of the zoo of algebraic modeling languages



CMPL, CPLEX Concert, GNU MathProg, Gurobi C++/Python API, linopy, MATLAB “problem-based optimization workflow”, Mosek Fusion, MOSEL, ompr, OPTMODEL, PuLP, Python-MIP, YALMIP, ZIMPL, ...

# What is JuMP?

## An algebraic modeling language in Julia

```
using JuMP, Ipopt
function constrained_linear_regression(A, y)
    m, n = size(A)
    model = Model(Ipopt.Optimizer)
    @variable(model, 0 <= x[1:n] <= 1)
    @variable(model, residuals[1:m])
    @constraint(model, residuals .== A * x .- y)
    @constraint(model, sum(x) == 1)
    @objective(model, Min, sum(residuals[i]^2 for i in 1:m))
    optimize!(model)
    print(solution_summary(model))
    return value.(x)
end
A, y = rand(30, 20), rand(30)
x = constrained_linear_regression(A, y)
```

# Who is (some of) JuMP?

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



Contributors 134



+ 123 contributors

# Why is JuMP interesting?

- Nice syntax
- Comprehensive documentation
- Vibrant community
- Open source
- Solver independent (35+ supported solvers)
- Embedded in Julia
- Supports interacting with solvers while they're running
- Low overhead for model generation
- Extensible to new solvers
- Extensible to new problem classes

# Why is JuMP interesting?

## Nice syntax

```
julia> @variable(model, x[i=1:2, j=1:2] >= 2i + j)
2×2 Matrix{VariableRef}:
 x[1,1]  x[1,2]
 x[2,1]  x[2,2]

julia> lower_bound.(x)
2×2 Matrix{Float64}:
 3.0  4.0
 5.0  6.0
```

```
julia> Y = [1 2; 2 1]
2×2 Matrix{Int64}:
 1  2
 2  1

julia> @constraint(model, X >= Y, PSDCone())
[X[1,1] - 1  X[1,2] - 2;
 X[2,1] - 2  X[2,2] - 1] ∈ PSDCone()
```

```
julia> @variable(model, x)
x

julia> @variable(model, y)
y

julia> @variable(model, a, Bin)
a

julia> @constraint(model, a == {x + y <= 1})
a == {x + y ≤ 1.0}
```

```
julia> @variable(model, x >= 0)
x

julia> @constraint(model, 2x - 1 ⊥ x)
[2 x - 1, x] ∈ MathOptInterface.Complements(2)
```

# Why is JuMP interesting?

## Embedded in Julia

```
square(x) = x^2
f(x, y) = (x - 1)^2 + (y - 2)^2

model = Model()

register(model, :square, 1, square; autodiff = true)
register(model, :my_f, 2, f; autodiff = true)

@variable(model, x[1:2] >= 0.5)
@NLOjective(model, Min, my_f(x[1], square(x[2])))
```

```
od_pairs = DBInterface.execute(
    db,
    """
    SELECT a.location as 'origin',
           b.location as 'destination'
    FROM locations a
    INNER JOIN locations b
    ON a.type = 'origin' AND b.type = 'destination'
    """,
)
```

```
SQLite.Query(SQLite.Stmt(SQLite.DB("/home/runner/work/JuMP.jl/JuMP
```

With a constraint that we cannot send more than 625 units between each pair:

```
for r in Tables.rows(od_pairs)
    @constraint(model, sum(x[r.origin, r.destination, :]) <= 625)
end
```

# Why is JuMP interesting?

## Low overhead for model generation

Model	# Variables	JuMP		Pyomo	GRB/C++
		direct_model	Model		
fac-25	67,651	1	1	4	0
fac-50	520,301	4	7	34	3
fac-75	1,732,951	14	20	128	8
fac-100	4,080,601	37	50	338	20
lqcp-500	251,501	2	1	24	3
lqcp-1000	1,003,001	7	11	107	10
lqcp-1500	2,254,501	17	23	234	23
lqcp-2000	4,006,001	32	50	417	40

**Table 1** Time (sec.) to generate each model and pass it to the solver. Models use Gurobi and terminate after a time limit of 0.0 seconds. `direct_model` uses the direct mode feature discussed in 3, and `Model` is the default way of constructing a JuMP model.



# JuMP 1.0.0 is released

releases · 24 Mar 2022

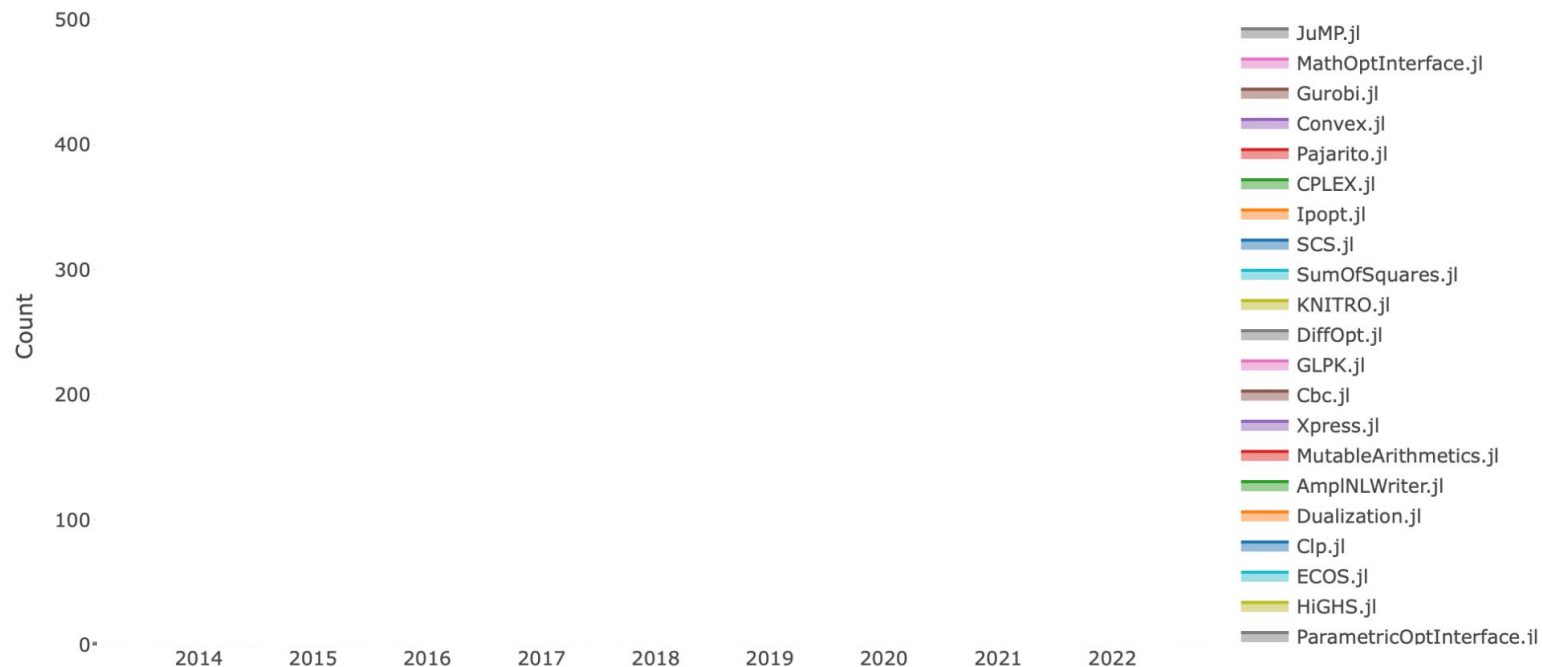
*Authors: Miles Lubin, Oscar Dowson, Iain Dunning, Joaquim Dias Garcia, Joey Huchette, and Benoît Legat.*

We are happy to announce the release of JuMP 1.0!

Nearly 10 years in the making, the release of JuMP 1.0 represents a major milestone in the history of JuMP. It also represents a stable platform from which we can continue to build new and exciting features for a long time into the future.

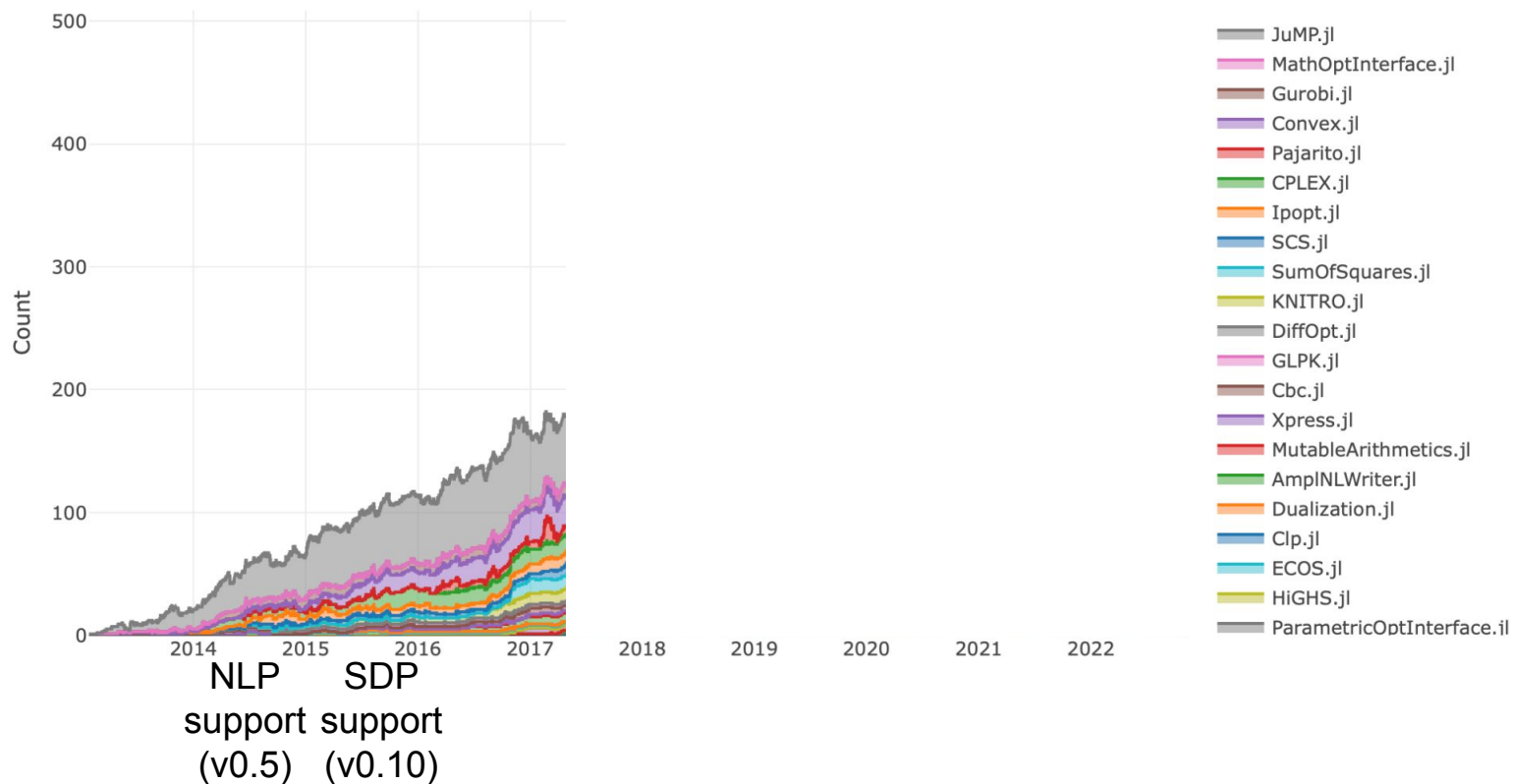
# Count of open issues by package

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



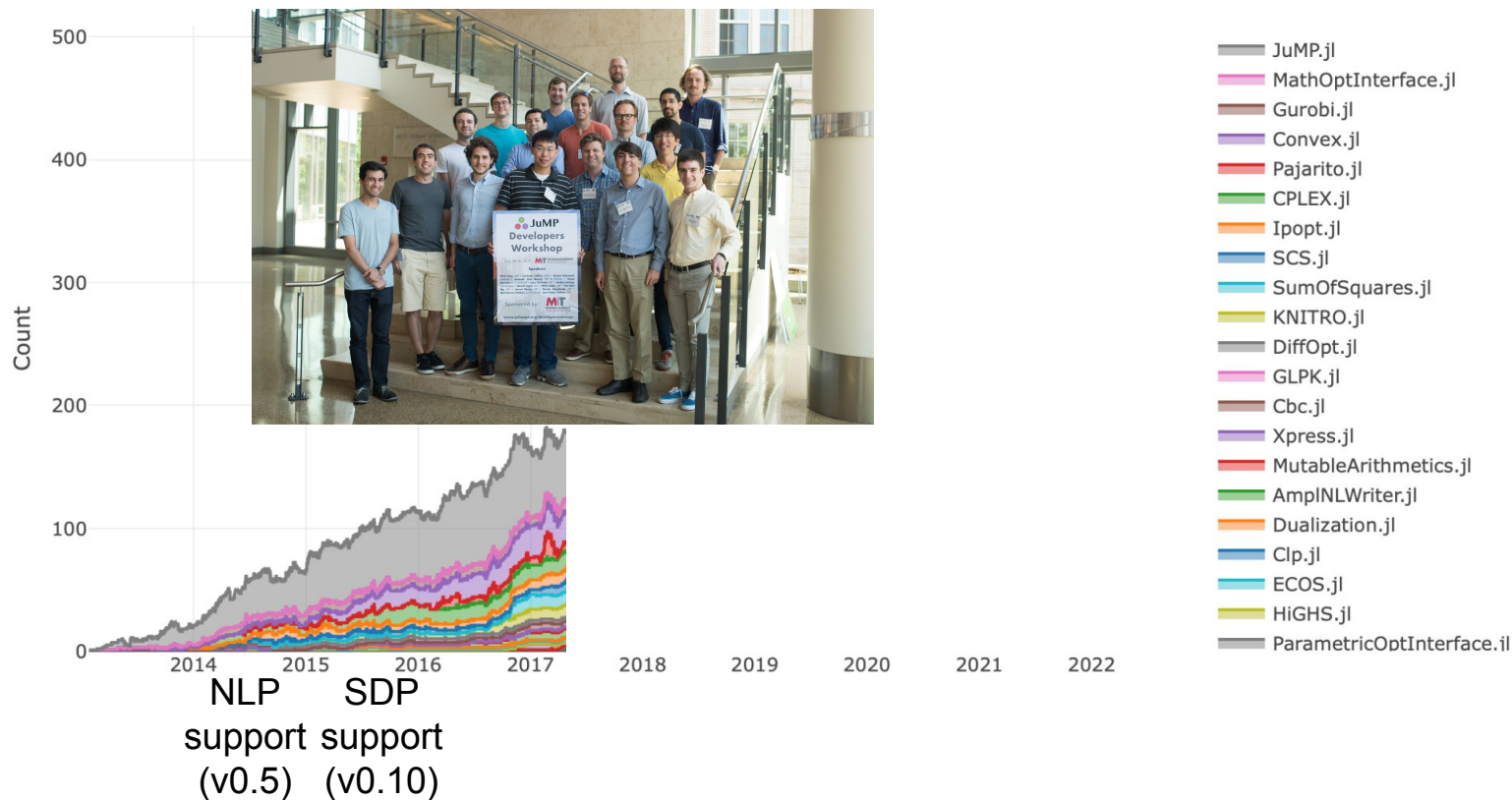
# Count of open issues by package

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



# Count of open issues by package

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

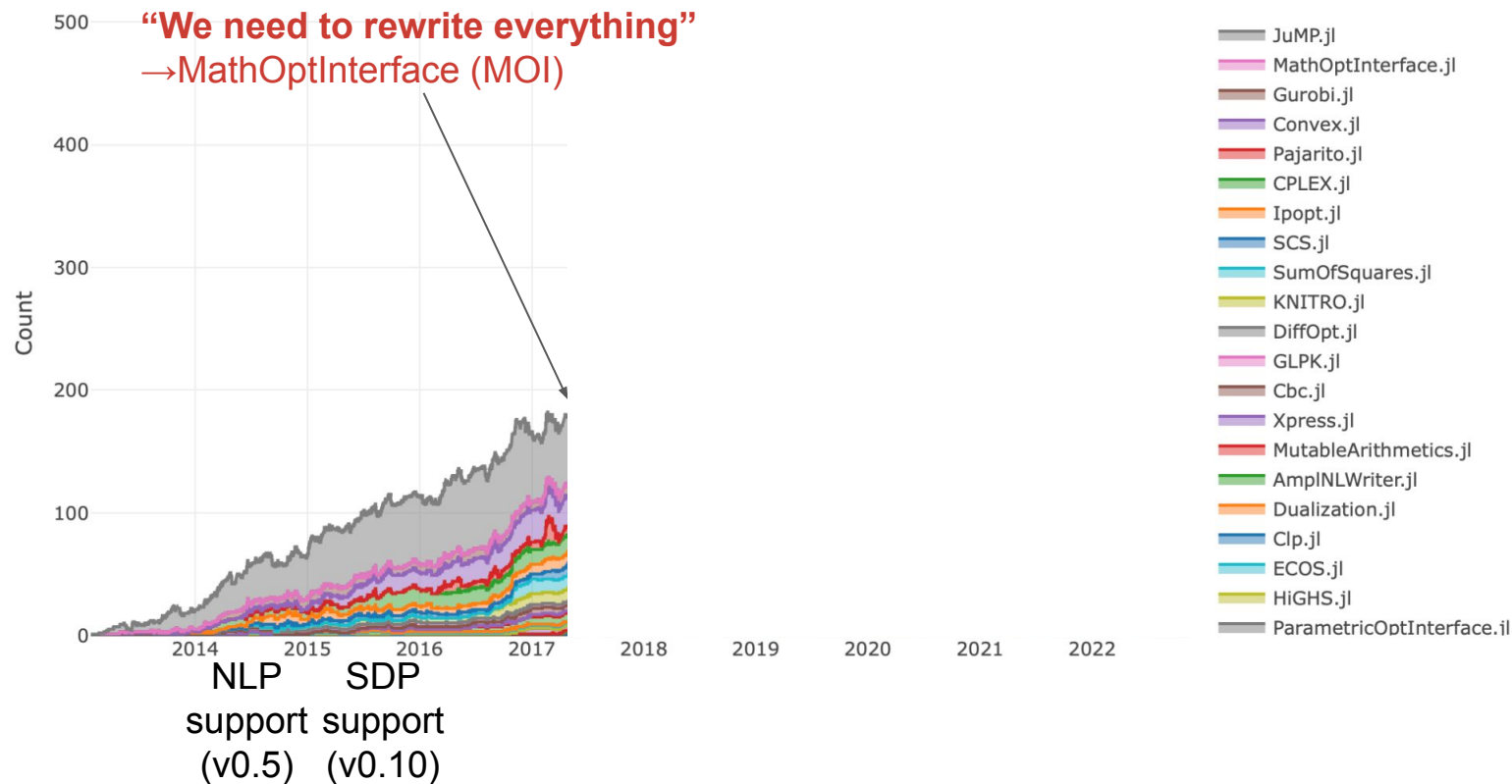


## How do I ...

- delete a constraint or variable?
- combine NLP constraints with conic constraints?
- add support for a new type of constraint?
- know if my incremental changes are passed efficiently to the solver?
- modify coefficients in the constraint matrix?
- provide a dual warm-start?
- access the irreducible inconsistent subsystem (IIS) from Gurobi?
- distinguish between a solver that stopped because of the time limit 1) *with* a solution and 2) *without*?
- check if a solution is feasible?
- access multiple solutions found by a solver?

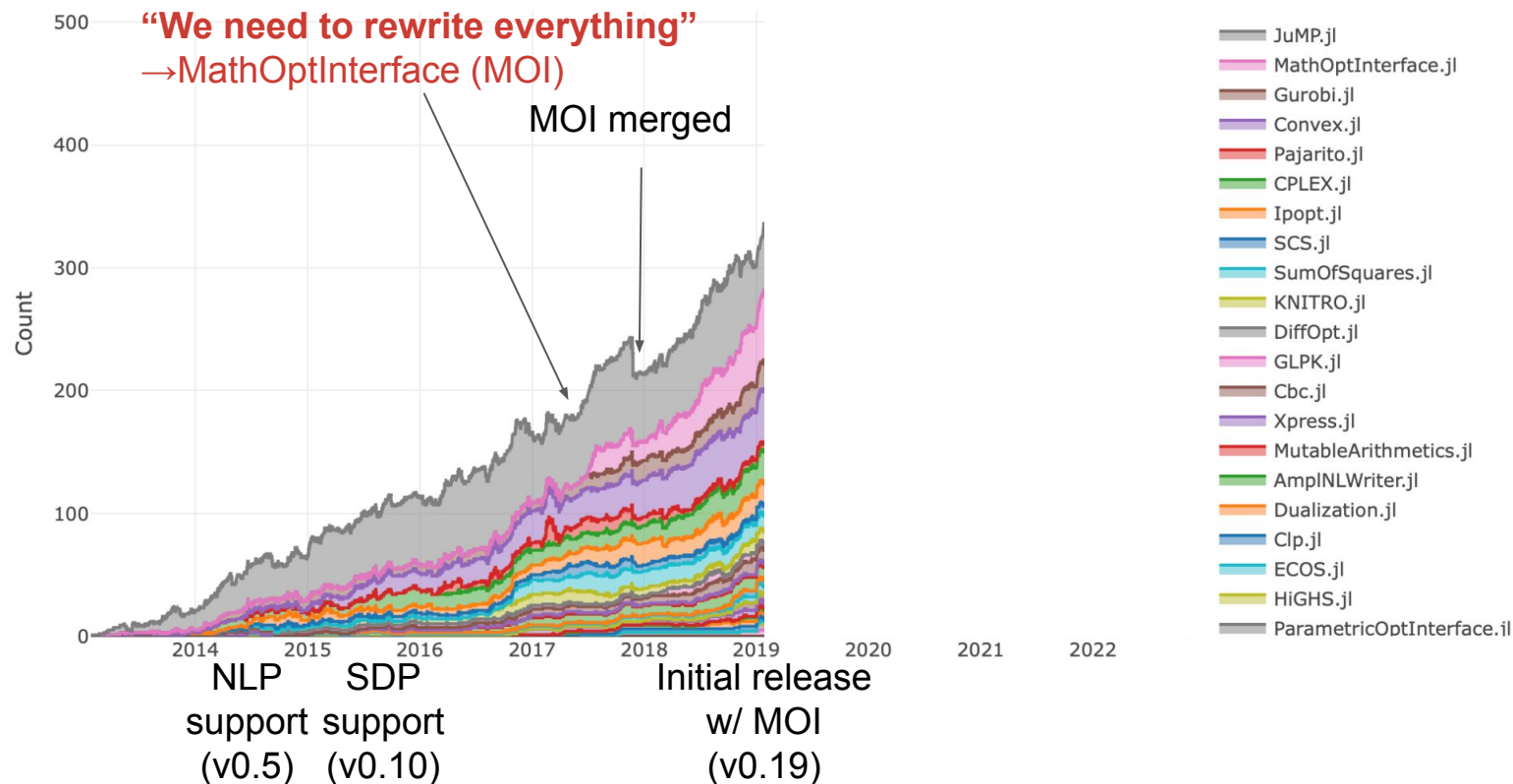
# Count of open issues by package

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



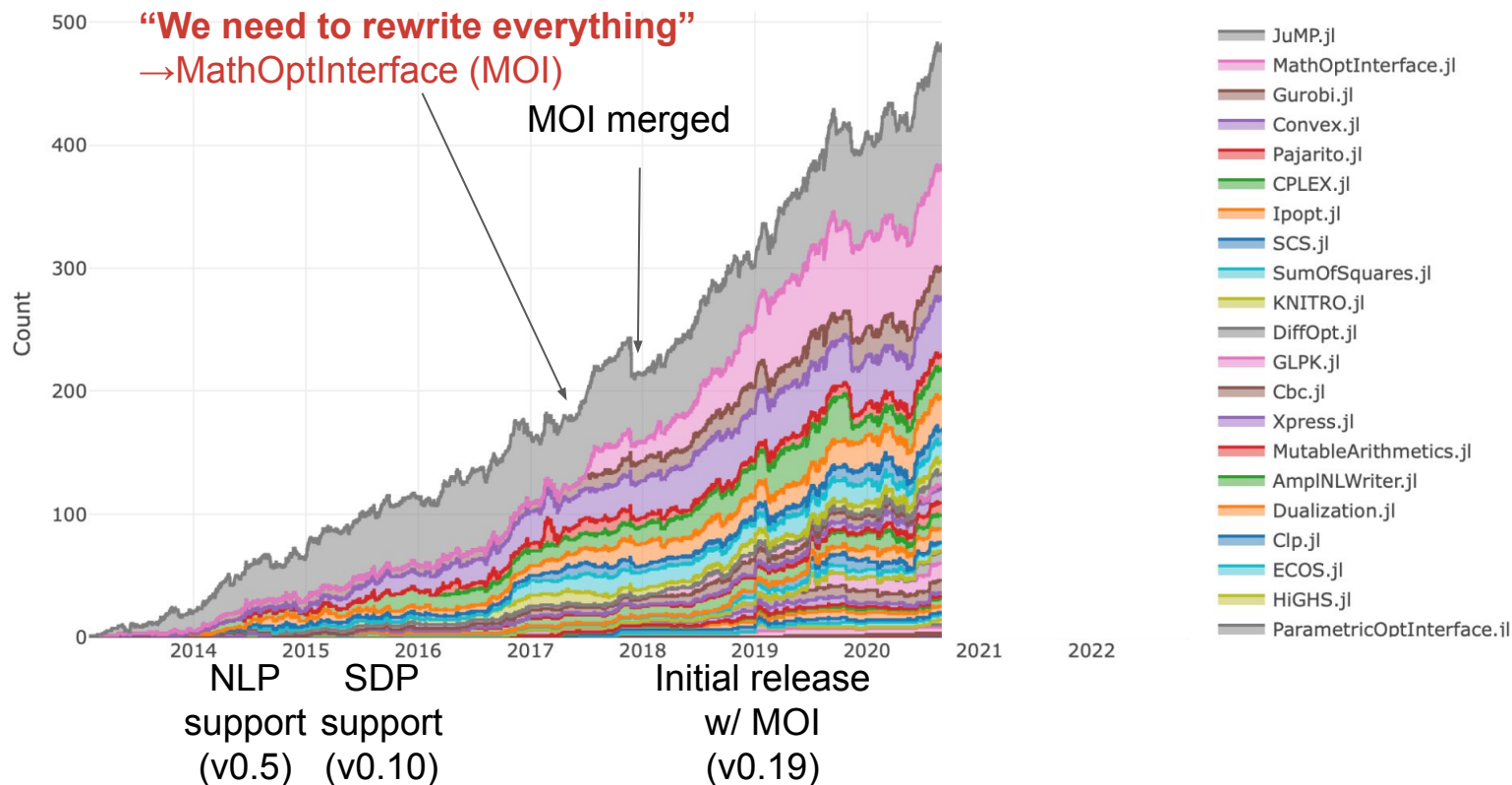
# Count of open issues by package

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



# Count of open issues by package

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

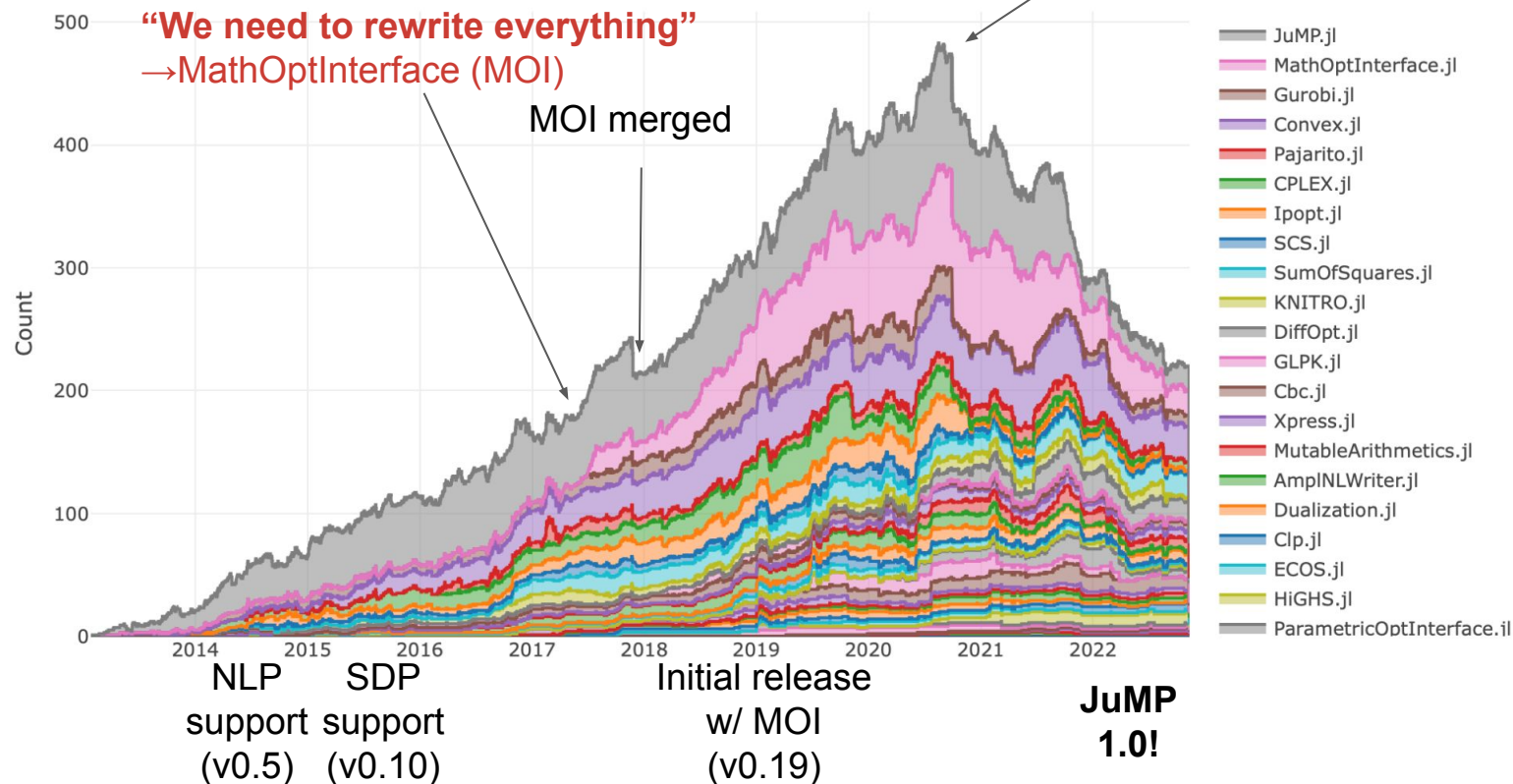




# Count of open issues by package

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

I started to get paid (NSF)



# Thank you!

Go to [jump.dev](https://jump.dev) for more information



Google Open Source



Northwestern | McCORMICK SCHOOL OF ENGINEERING



And the whole JuMP community!