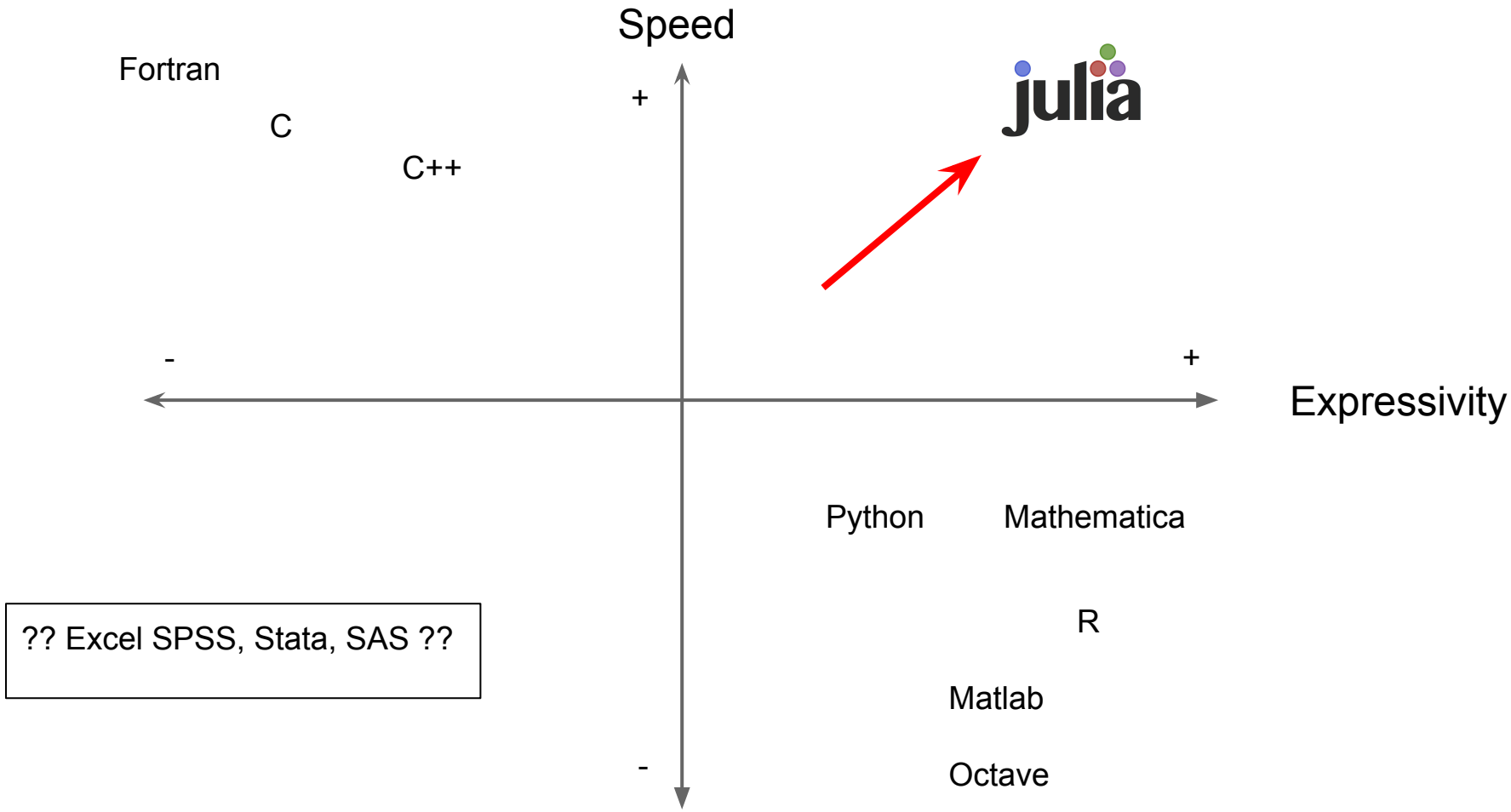
Why julia

Perl for Numerical Computing

sed	cat	patch
awk	csplit	sort
grep	cut	tr
	diff	tsort
	expand	uniq
	paste	wc

Numeric Computing Environments

Fortran	R
C, C++	Stata
Python, Numpy, Scipy	SPSS
Matlab	SAS
Mathematica	Maple
Octave	Excel


Speed
Fortran
C
C++
julia
+
-
+
-
Expressivity
Python
Mathematica
R
Matlab
Octave
?? Excel SPSS, Stata, SAS ??

Expressivity

Functions

```
f(x) = x + 2x - 4x^3

function count_whitespace(text)
    white = [' ', '\t', '\n']
    sum([c in white ? 1 : 0 for c=text])
end

map(x -> x^2, [1 2 3 4])
```

Arrays

```
x = [2 4 1
     4 8 1
     1 2 1]

a = [0.5, -0.25, 1]

x * a
```

```
3-element Array
{Float64,1}:
 1.0
 1.0
 1.0
```

Type System

```
type Point
    x::Real
    y::Real
end
```

```
type Rectangle
    top_left::Point
    bottom_right::Point
end

type Circle
    center::Point
    radius::Real
end
```

Multiple Dispatch

```
-(p1::Point, p2::Point) = (p1.x-p2.x, p1.y-p2.y)

area(circ::Circle) = pi * circ.radius^2

function area(rect::Rectangle)
    abs(prod(rect.bottom_right - rect.top_left))
end
```

Calling External Code

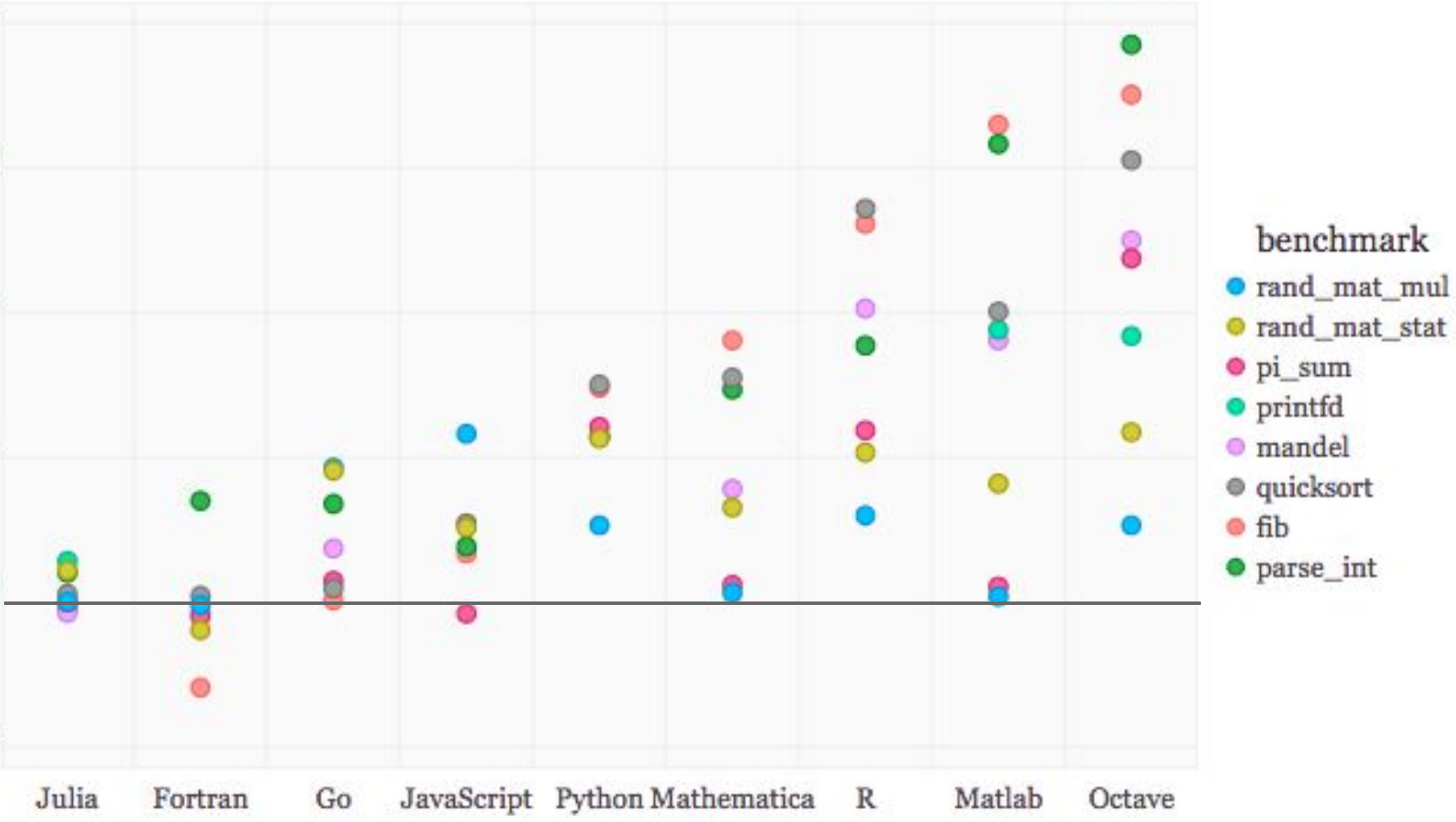
```
time = ccall((:clock, "libc"), Int32, ())

@pyimport scipy.optimize as so

so.newton(x -> cos(x) - x, 1)

dir = "/Users/erik"
readchomp(    `find $dir -exec wc {} \;`
           |> `sort -r -k1,1`)
```

Speed


benchmark
rand_mat_mul
rand_mat_stat
pi_sum
printfd
mandel
quicksort
fib
parse_int
Julia
Fortran
Go
JavaScript
Python
Mathematica
R
Matlab
Octave

	Fortran gcc 4.8.1	Julia 0.2	Python 2.7.3	R 3.0.2	Matlab R2012a	Octave 3.6.4	Mathematica 8.0	JavaScript V8 3.7.12.22	Go go1
fib	0.26	0.91	30.37	411.36	1992.00	3211.81	64.46	2.18	1.03
parse_int	5.03	1.60	13.95	59.40	1463.16	7109.85	29.54	2.43	4.79
quicksort	1.11	1.14	31.98	524.29	101.84	1132.04	35.74	3.51	1.25
mandel	0.86	0.85	14.19	106.97	64.58	316.95	6.07	3.49	2.36
pi_sum	0.80	1.00	16.33	15.42	1.29	237.41	1.32	0.84	1.41
rand_mat_stat	0.64	1.66	13.52	10.84	6.61	14.98	4.52	3.28	8.12
rand_mat_mul	0.96	1.01	3.41	3.98	1.10	3.41	1.16	14.60	8.51

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

Fast
Expressive
Numerical Computing

julialang.org