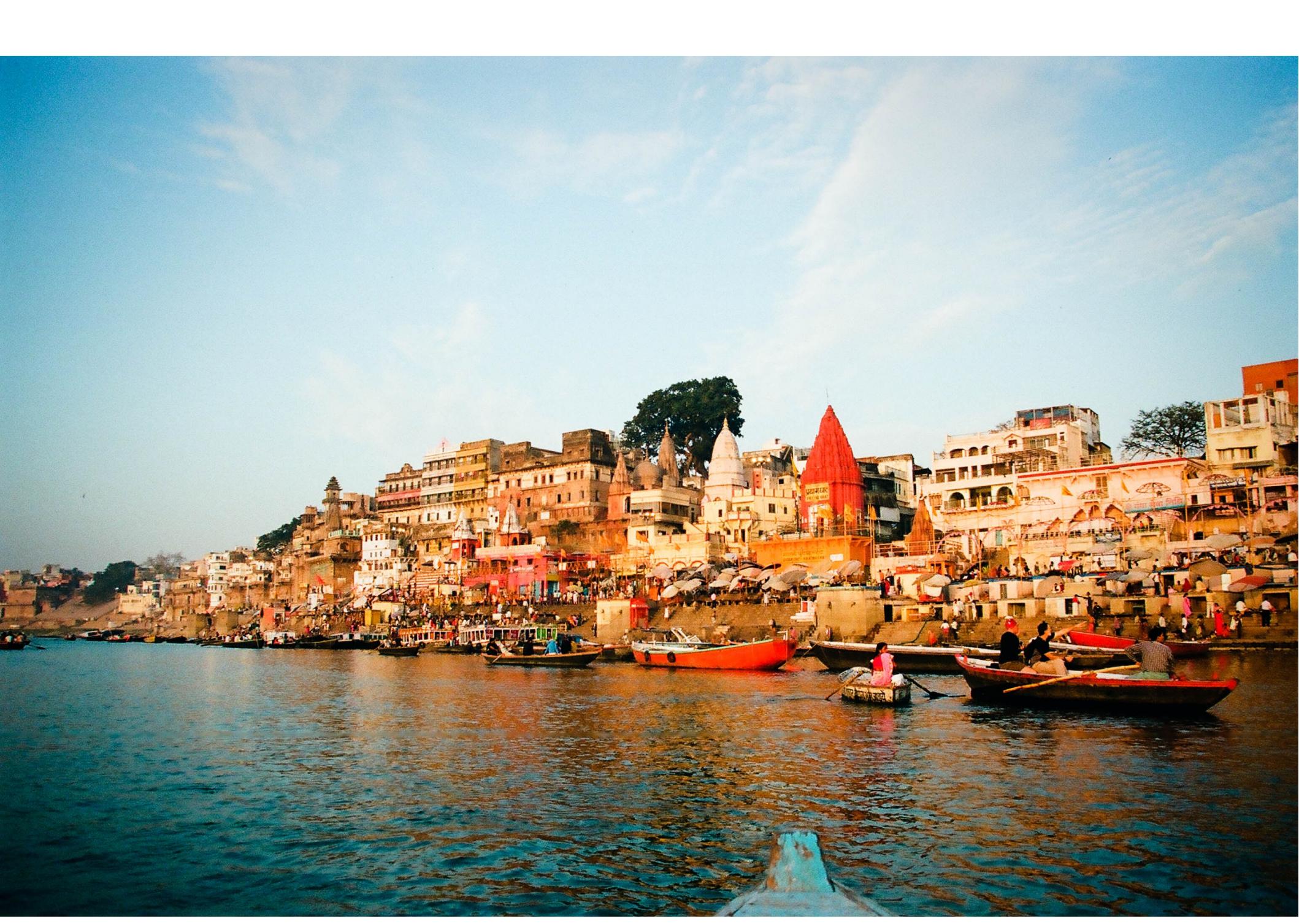


namaste



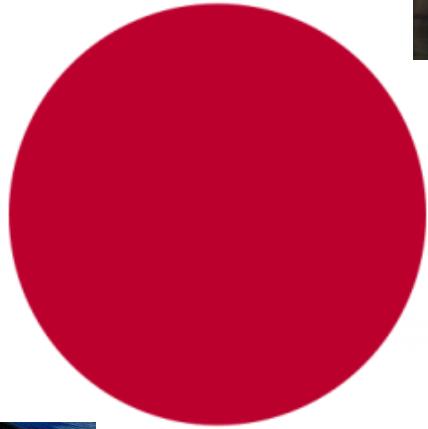


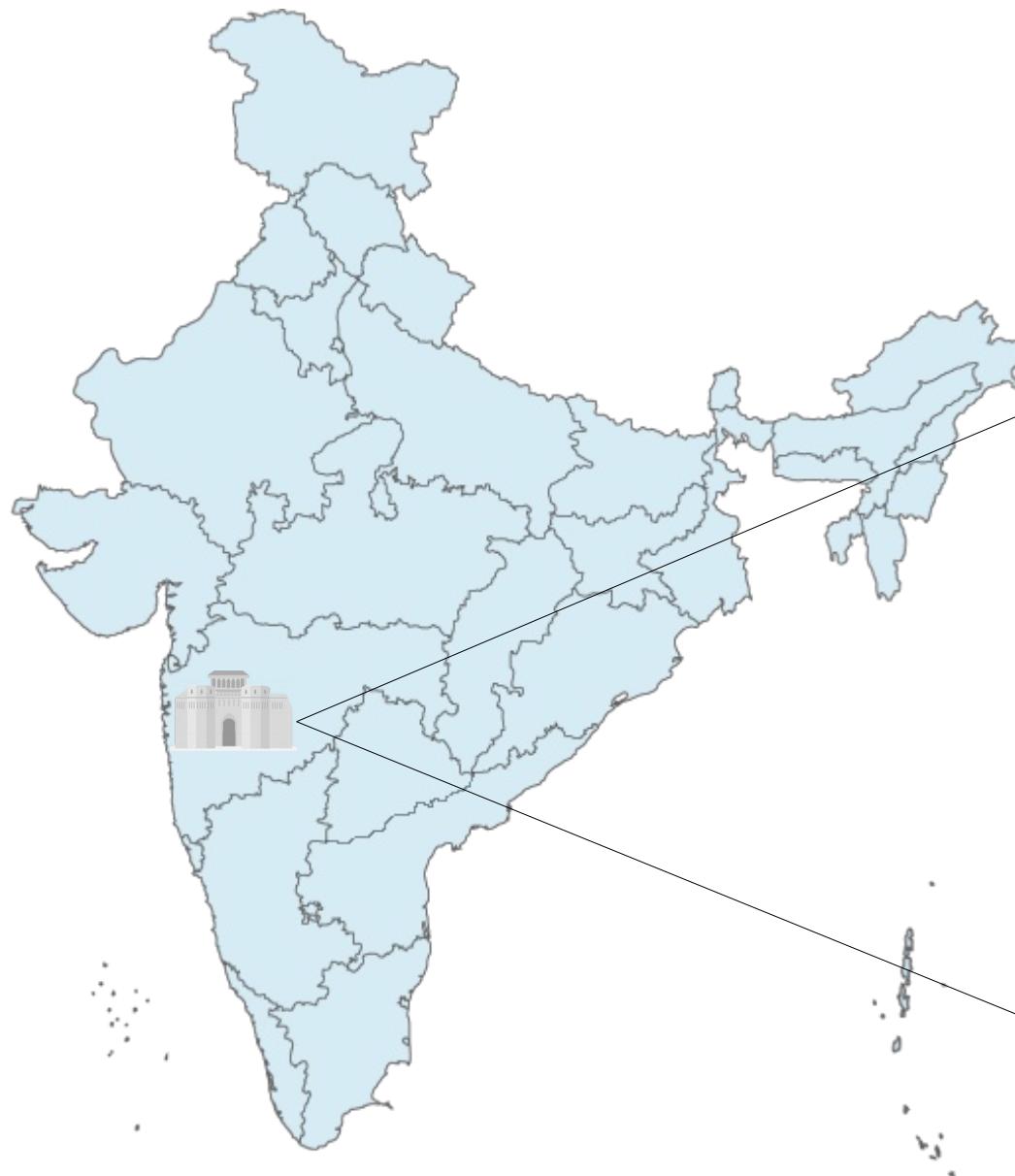




# What India Dreams

India must master Western science and  
yet preserve its Culture and Heritage.





City of Pune.

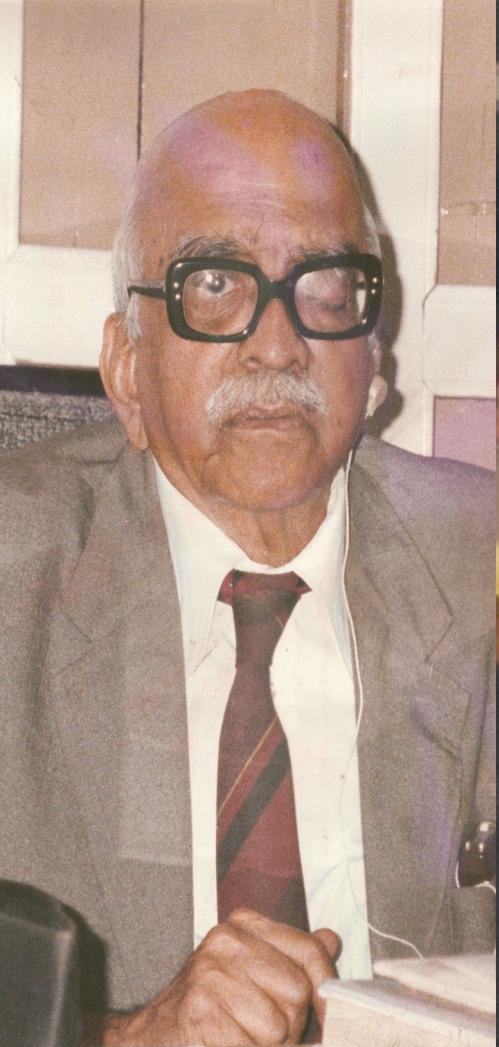
Population: 6 million.  
Oxford of the East.

# Sameer Deshmukh

 [github.com/v0dro](https://github.com/v0dro)

 @v0dro





Dr. Gopal  
Deshmukh



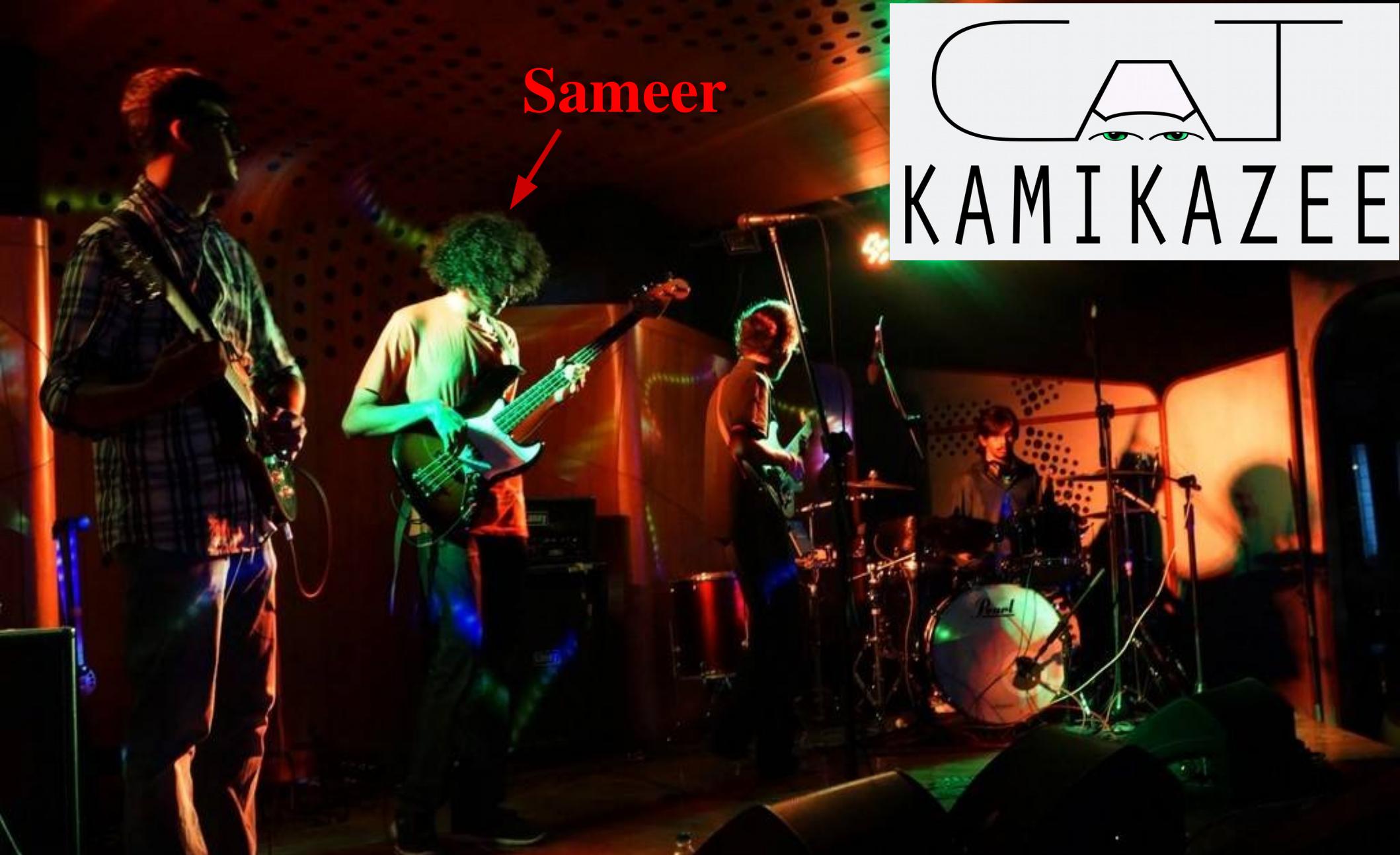
Dr. Hemchandra  
Deshmukh



Dr. Satish  
Deshmukh



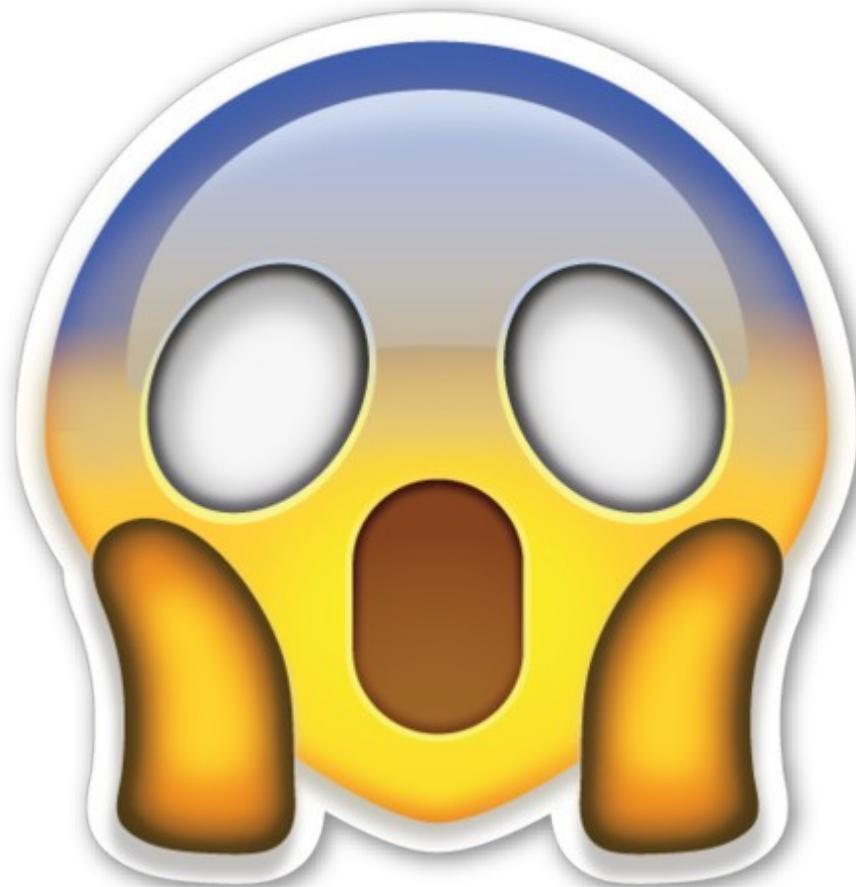
Sameer  
Desmukh



Sameer



[www.soundcloud.com/catkamikazee](https://www.soundcloud.com/catkamikazee)





Me

# Pune Ruby Users Group



@punerb



@punerb

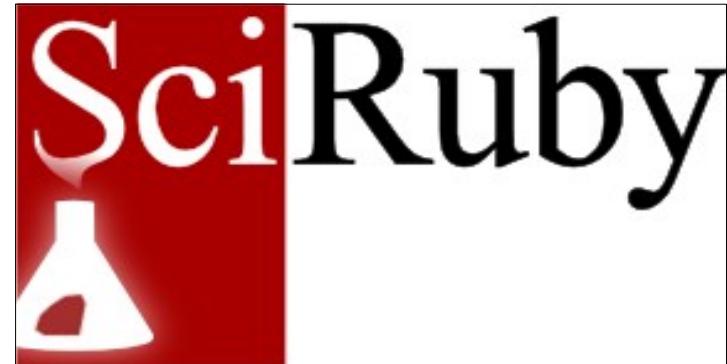
[www.punerb.org](http://www.punerb.org)



@deccanrubyconf

[www.deccanrubyconf.org](http://www.deccanrubyconf.org)

# Ruby Science Foundation



 @sciruby  
 @sciruby

[www.sciruby.com](http://www.sciruby.com)



# Data Analysis in Ruby with daru

# daru

## (Data Analysis in RUbY)

daru

दारु

(Hindi)

— —



sake

alcohol

library for  
analysis, cleaning, manipulation and  
visualization  
of data.

# Advanced Data indexing

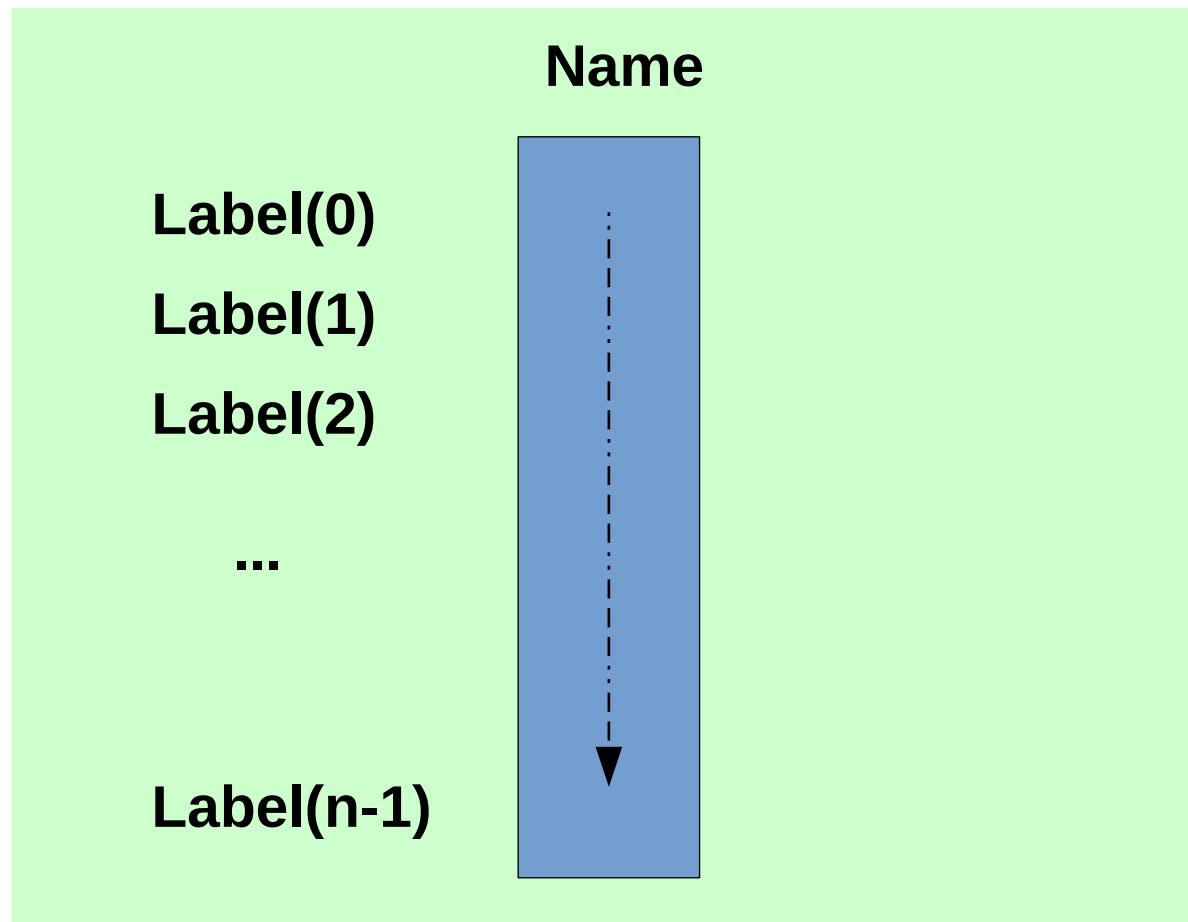
Read/write many data sources

Works well with 'wild' data

Ephemeral statistics functions

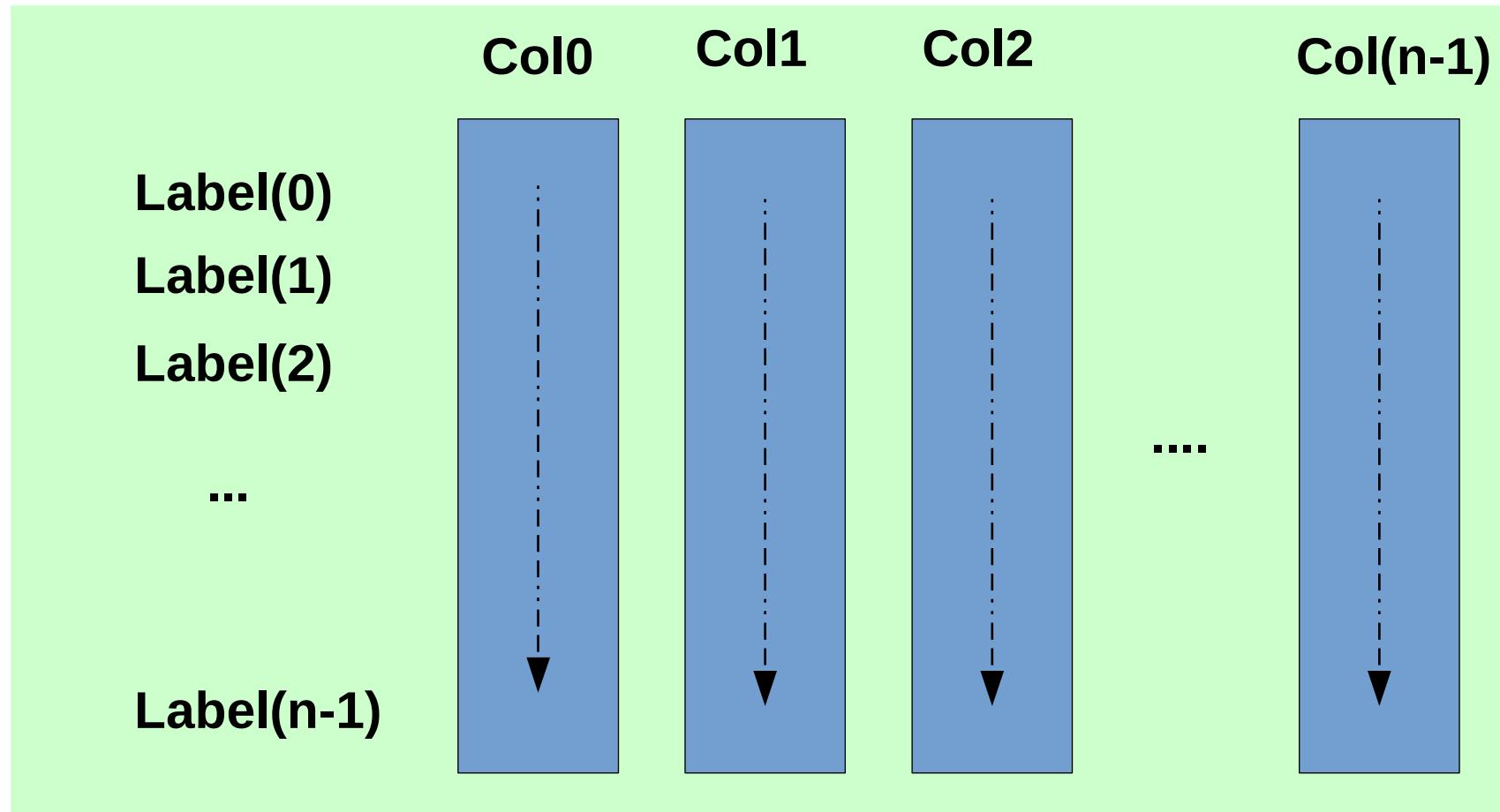
# Daru::Vector

Heterogenous Array that can be indexed on any Ruby object.



# Daru::DataFrame

2D spreadsheet like data structure indexed by rows or columns.



# Data visualization with Nyaplot, GNUMplotrb and Gruff.

iruby notebook  
gem install iruby

Browser based Ruby REPL  
for interactive computing.

A screenshot of a Jupyter Notebook interface. At the top, the URL bar shows "localhost:8888/notebooks/Untitled2.ipynb?kernel\_name=ruby". Below it is a browser toolbar with various icons. The main title is "jupyter Untitled2 Last Checkpoint: 02/07/2016 (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, and Help. The toolbar below has buttons for New, Open, Save, Cell Type, Cell Selection, Cell Execution, Cell Kernel, and Cell Toolbar dropdowns set to "None". The notebook content area shows an input cell (In [1]) containing Ruby code to require 'matrix' and create a 3x3 matrix. The output cell (Out[1]) displays the resulting 3x3 matrix with values 1, 2, 3 in each row. Three red arrows point from the explanatory text below to the output cell, the input cell, and the kernel name in the URL bar respectively.

```
In [1]: require 'matrix'  
Matrix[[1,2,3], [1,2,3], [1,2,3]]
```

```
Out[1]: 
$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

```

Output cell – can  
render HTML/CSS/JS

Input cell – accepts  
Ruby code

Runs in your  
browser

**60%** of a data analyst's time is spent on  
cleaning data.

# Acts as glue between other SciRuby libraries

- `statsample` for Statistics.
- `mixed_models` for Mixed Models.
- `daru-td` for Treasure Data.
- `nmatrix` for efficient data storage.

# statsample-glm

```
gem install statsample-glm
```

Logistic, probit, poisson, normal  
regression methods in Ruby.

Provides an R-like formula language  
for specifying regressions.

“Y ~ a+a:b+c+c:d”

$$Y = \beta_0 + a * \beta_1 + a * b * \beta_2 + c * \beta_3 + c * d * \beta_4$$

# Use Case:

# Kaggle Animal Shelter Data



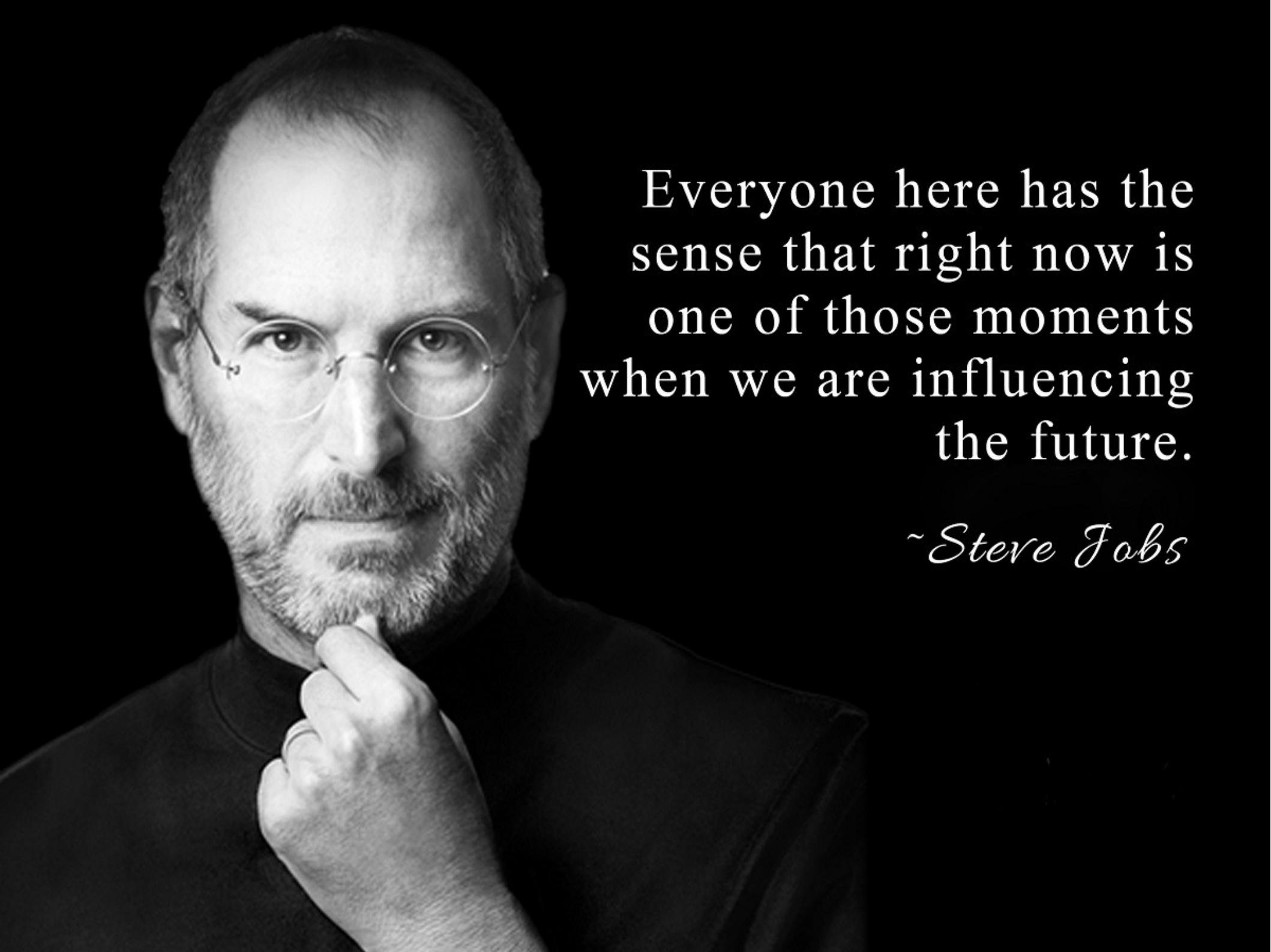
OMG I've had too  
much daru!!

STOP! STOP!

# New Ideas for better Ruby

“Any sufficiently advanced  
technology is indistinguishable  
from magic.”

- Arthur C. Clarke

A black and white close-up portrait of Steve Jobs. He is looking slightly to the left of the camera with a thoughtful expression, his right hand resting against his chin. He has short, light-colored hair, wears round-rimmed glasses, and has a full, light-colored beard and mustache.

Everyone here has the  
sense that right now is  
one of those moments  
when we are influencing  
the future.

*~Steve Jobs*

# Writing C extensions

- FFI gem.
- Rice.
- SWIG.
- Writing C bindings manually.



Rubyist!  
Write me a C extension!

```
def factorial n
  n > 1 ? n*factorial(n-1) : 1
end
```

```
unsigned long long int
calc_factorial(unsigned long long int n)
{
    return (n > 1 ? n*calc_factorial(n-1) : 1);
}

static VALUE
cffactorial(VALUE self, VALUE n)
{
    return ULL2FIX(
        calc_factorial(NUM2ULL(n)));
}
```

```
void Init_factorial()
{
    VALUE cFact = rb_define_class("Fact",
rb_cObject);

    rb_define_method(cFact, "factorial",
cfactorial, 1);
}
```

```
a = Fact.new  
a.factorial(8000)
```

# Big Problems

- Difficult and irritating to write.
- Time consuming to debug.
- Tough to trace memory leaks.
- Change mindset from high level to low level language.
- Need to care about small things.<sup>TM\*</sup>

\*Matz – Keynote at Red Dot Ruby Conf 2016, Singapore.

# Rubex

v0dro / rubex

Unwatch ▾ 1

Star 4

Fork 0

Code

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

Settings

rubex - A Crystal-inspired language for writing Ruby extensions. — Edit

4 commits

1 branch

0 releases

1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

Rubex is a Crystal-inspired superset of  
Ruby that compiles to C.

```
class Fact

def factorial(unsigned long long int n)
    n > 1 ? n*factorial(n-1) : 1
end

end
```

# Create a C static array and return a Ruby Array

```
def adder(n)
```

```
    a = StaticArray(i32, n)
```

```
    i32 i = 0
```

```
    i32 sum = 0
```

```
    a.each(n) { a[i] = i*5 }
```

```
    for 0 <= i < n do
```

```
        sum += a[i]
```

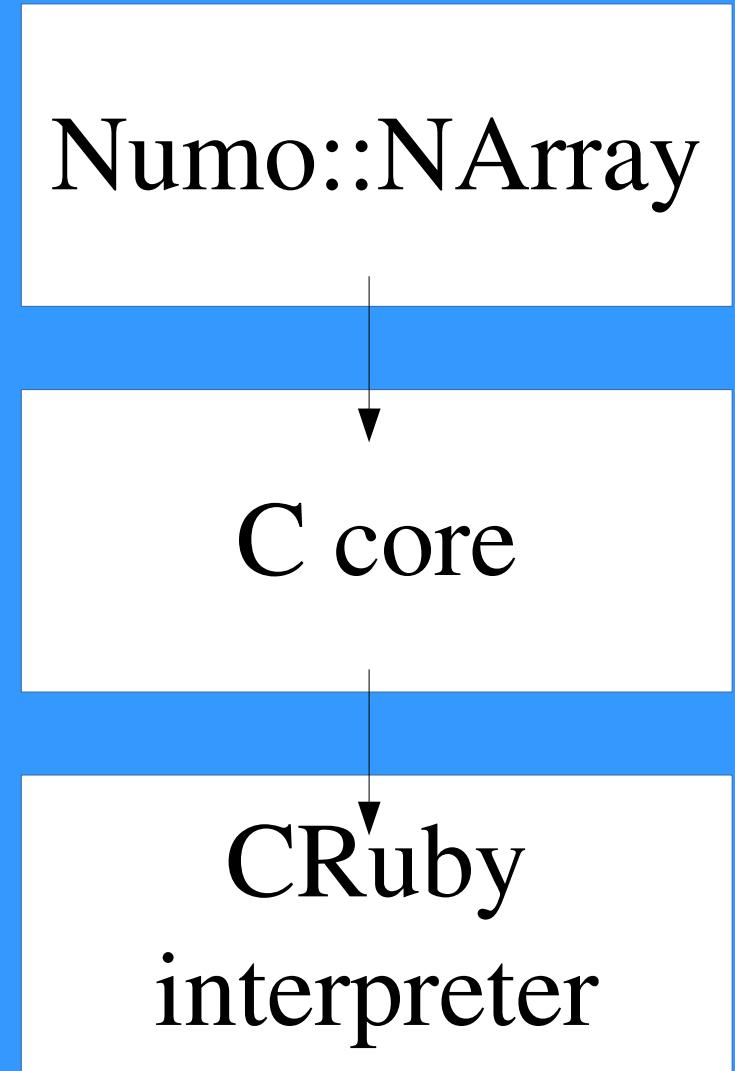
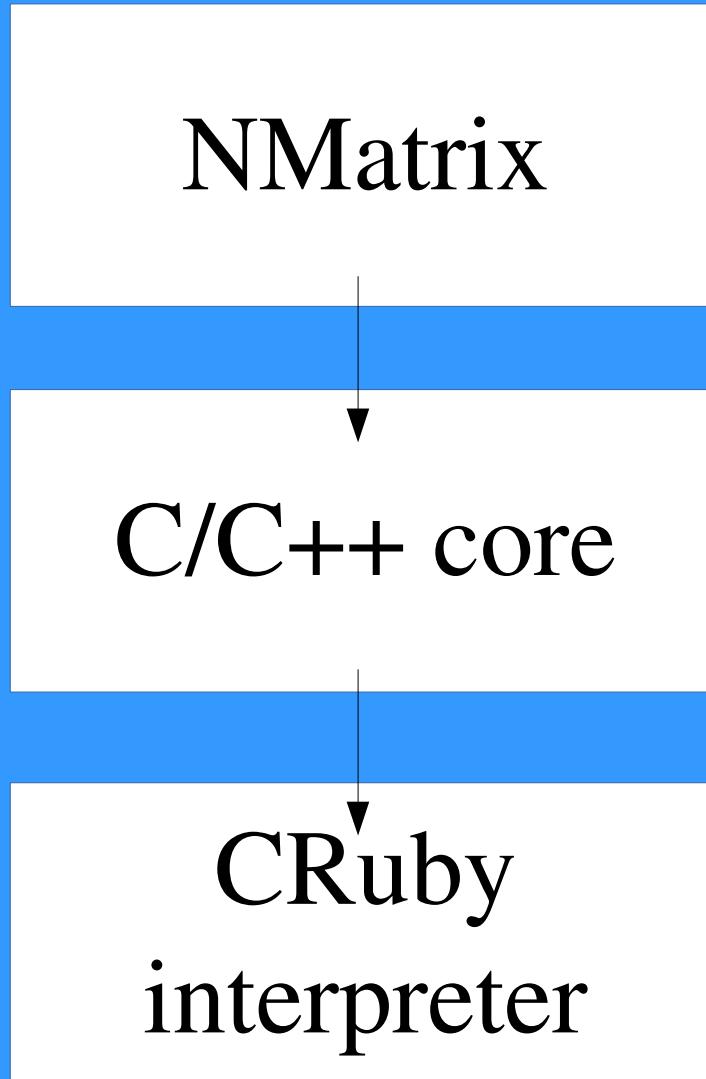
```
    end
```

```
    sum
```

```
end
```

<https://github.com/v0dro/rubex>

# Scientific Computing on JRuby



JRuby backend for the NMatrix  
Ruby API –  
Sci. Computing on JVM.

Uses Apache Commons Math library for storage and operations on internal Java arrays.

Allows interfacing JRuby libraries with jBLAS for performance.

[https://github.com/prasunanand/nmatrix/tree/jruby\\_port](https://github.com/prasunanand/nmatrix/tree/jruby_port)

# Symbolic Computation in Ruby with `symengine.rb`

$$(x - y) * (x ** y / z)$$

```
require 'symengine'
```

```
x = SymEngine::Symbol.new("x")
```

```
y = SymEngine::Symbol.new("y")
```

```
z = SymEngine::Symbol.new("z")
```

```
f = (x - y) * (x ** y / z)
```

```
f.expand.to_s
```

```
# x**(1 + y)/z - x**y*y/z
```

```
f == - (x**y*y/z) + (x**y*x/z)
```

```
# true
```

[https://github.com/symengine/  
symengine.rb](https://github.com/symengine/symengine.rb)

# Ruby in Space

NASA SPICE

Ruby wrapper `spice_rub`

```
require 'spice_rub'

k_pool = SpiceRub::KernelPool.instance
k_pool.load_folder("spec/data/kernels")

epoch = SpiceRub::Time.now
moon = SpiceRub::Body.new(:moon)
earth = SpiceRub::Body.now(:earth)

earth.position_at(epoch)
moon.distance_from(:earth, epoch)
# 395791.1464913574 (Km)
```

[https://github.com/gau27/spice\\_rub](https://github.com/gau27/spice_rub)

# Cool SciRuby Stickers



# Acknowledgements

- **@agisga** and **@lokeshh** for statistics with `daru` and `statsample-glm`.
- **@gau27** for `spice_rub`.
- **@prasunanand** for `NMatrix` on JRuby.
- **@rajithv** for `symengine.rb`.
- **@gnilrets**, **@mrkn**, **@zverok** and all the other contributors to `daru`.

Thank You  
Ruby Kaigi



Any questions?