# Software Carpentry: Lessons Learned

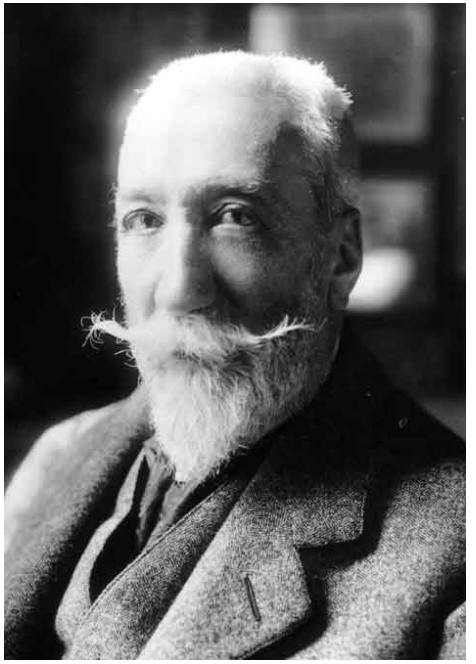
Greg Wilson

software carpentry

# Then

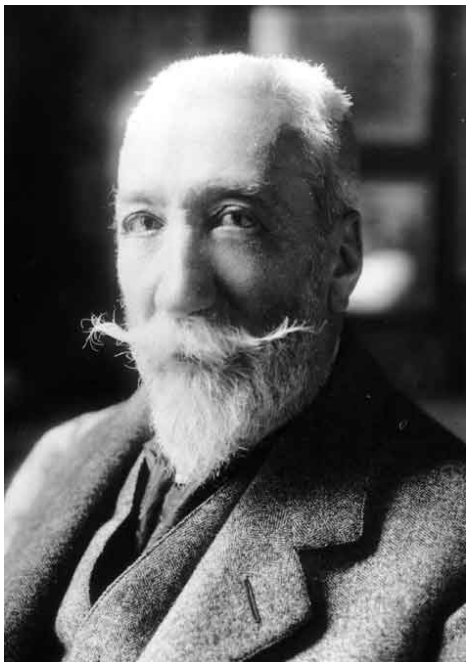
Anatole France (1844-1924)

“The law, in its majestic equality, forbids the rich and poor alike to sleep under bridges, to beg in the streets, and to steal bread.”



# Now

Anatole France (1844-1924)



“The law, in its majestic equality, forbids the rich and poor alike to sleep under bridges, to beg in the streets, and to steal bread.”

Today, thanks to computers,
every scientist can devote her working life
to getting software installed.

5-15%

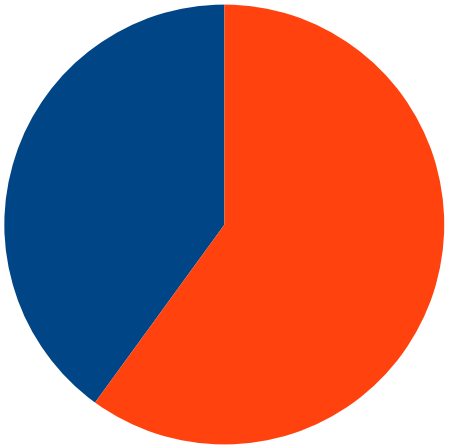
GPU clusters to analyze petabytes in the cloud

85-95%

Sending each other spreadsheets by email

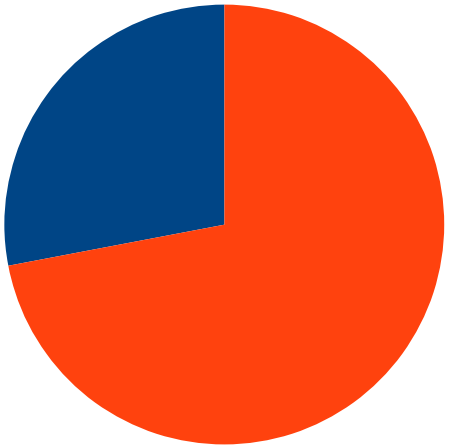
# Surely You're Exaggerating

1. How many graduate students write shell scripts to analyze each new data set instead of running those analyses by hand?



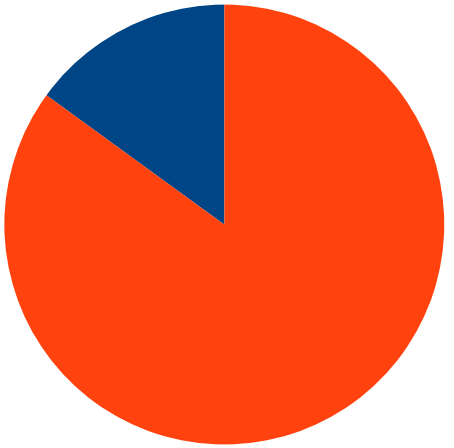
# Surely You're Exaggerating

2. How many of them use version control to keep track of their work and collaborate with colleagues?



# Surely You're Exaggerating

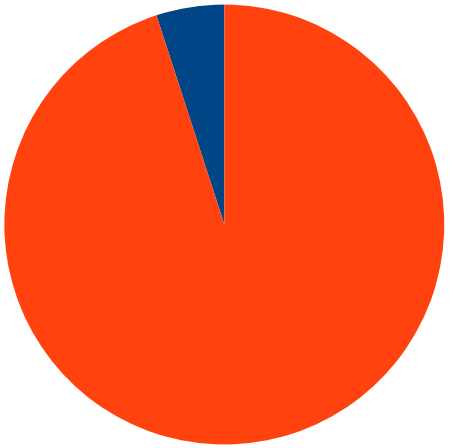
3. How many routinely break large problems into pieces small enough to be
   - comprehensible,
   - testable, and
   - reusable?



# Surely You're Exaggerating

3. How many routinely break large problems into pieces small enough to be
   - comprehensible,
   - testable, and
   - reusable?

   And how many know those are the same things?



# Where Are Your Goalposts?

A *computationally competent* scientist can:

- Manage and process data
- Tell if it's been processed correctly
- Find and fix problems when it hasn't been
- Keep track of what she has done
- Share her work with others

*Efficiently*

# It Is Therefore Obvious That...

- Put more computing courses in the curriculum!

- But it's already full

# It Is Therefore Obvious That...

- Put a little computing in every course!
  - Still adds up: 5 minutes/lecture = 4 courses/degree
  - First thing cut when running late

# It Is Therefore Obvious That...

- And no matter what we do...


- The blind leading the blind


software
carpentry

*If you build a man a fire,*
*you'll keep him warm for a night.*
*If you set a man on fire,*
*you'll keep him warm for the rest of his life.*

— Terry Pratchett

# What We Teach

Unix shell

Version control

Python/R/MATLAB

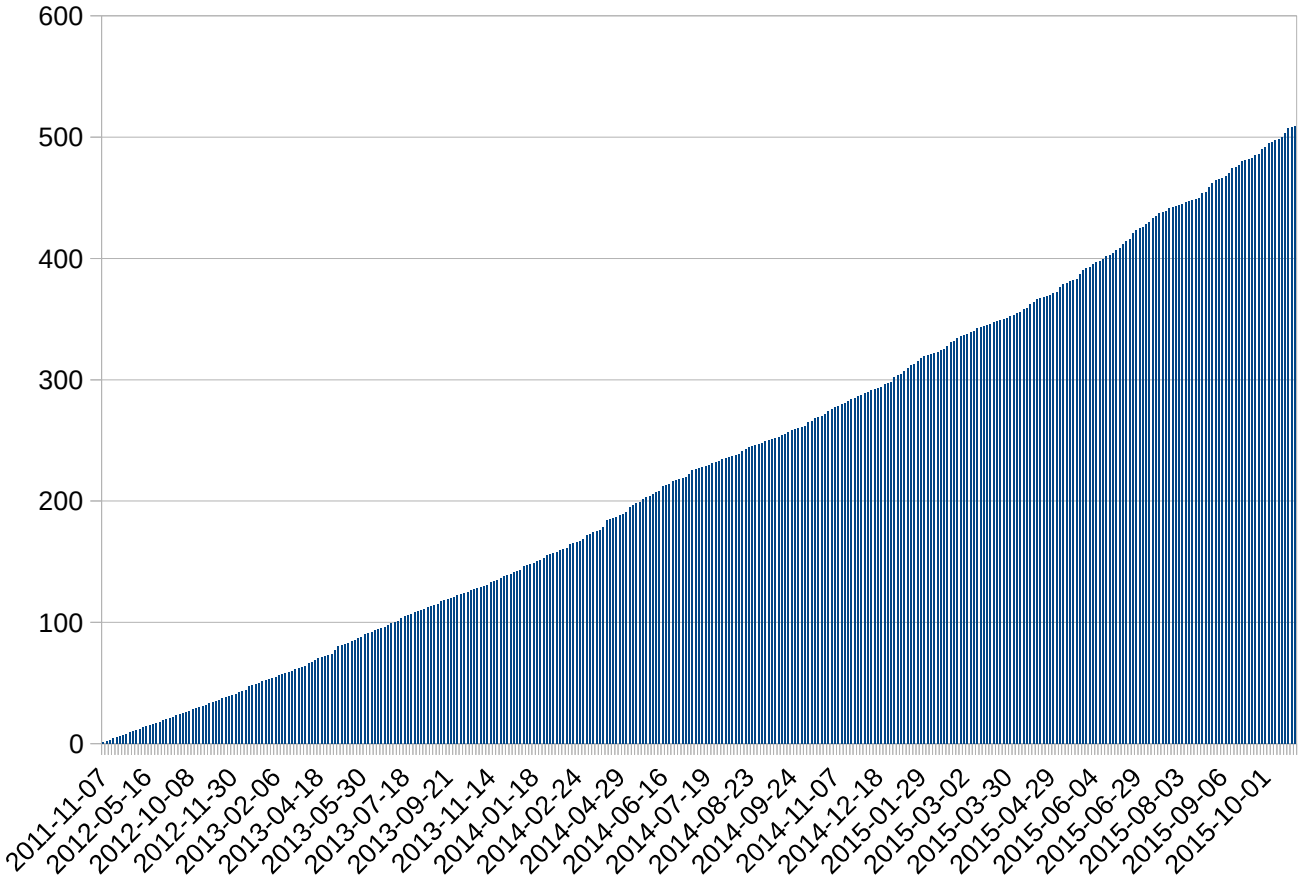
SQL

Make

# What We *Actually* Teach

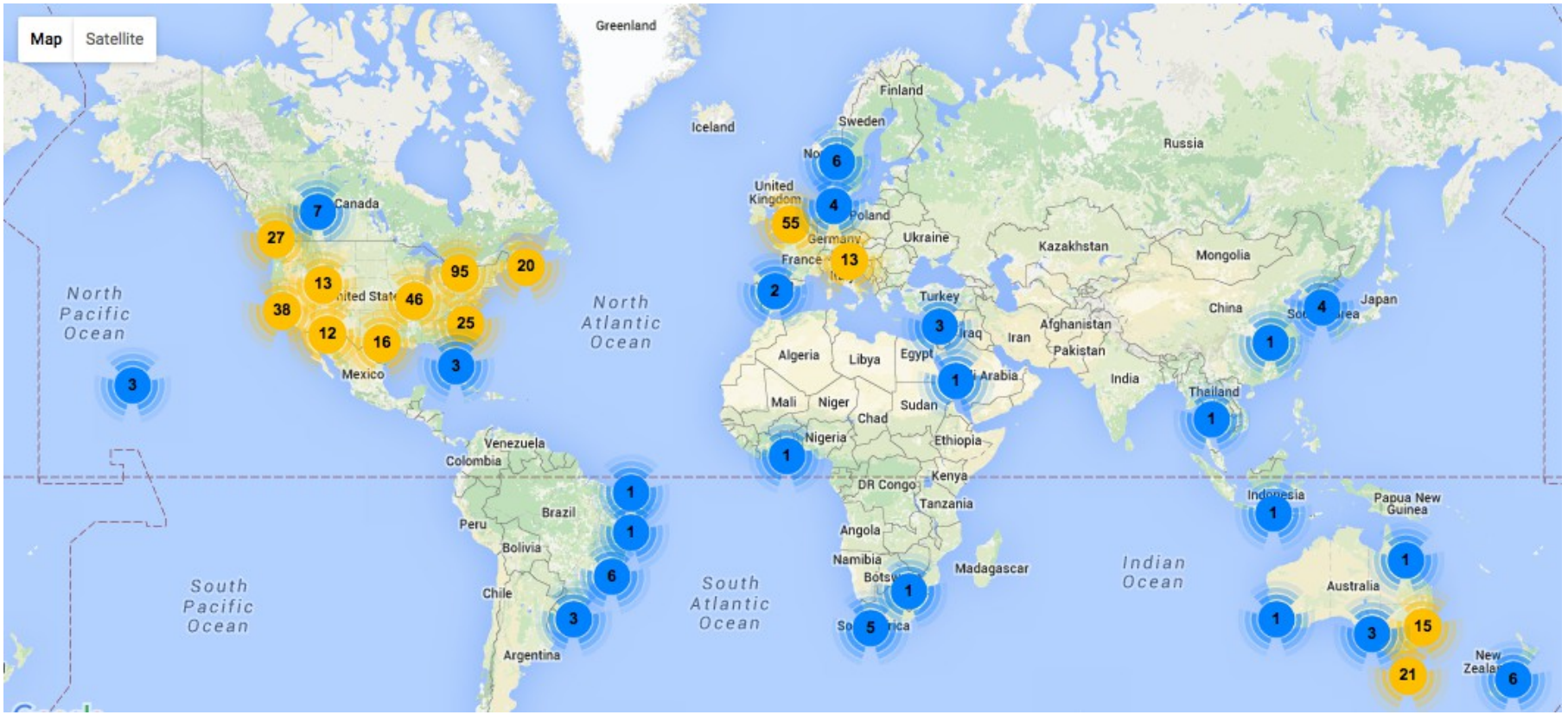
| | | |
|---|---|---|
| Unix shell | => | Task automation |
| Version control | => | Track and share work |
| Python/R/MATLAB | => | Modular programming |
| SQL | => | Data management |
| Make | => | Reproducibility |

# How It's Going



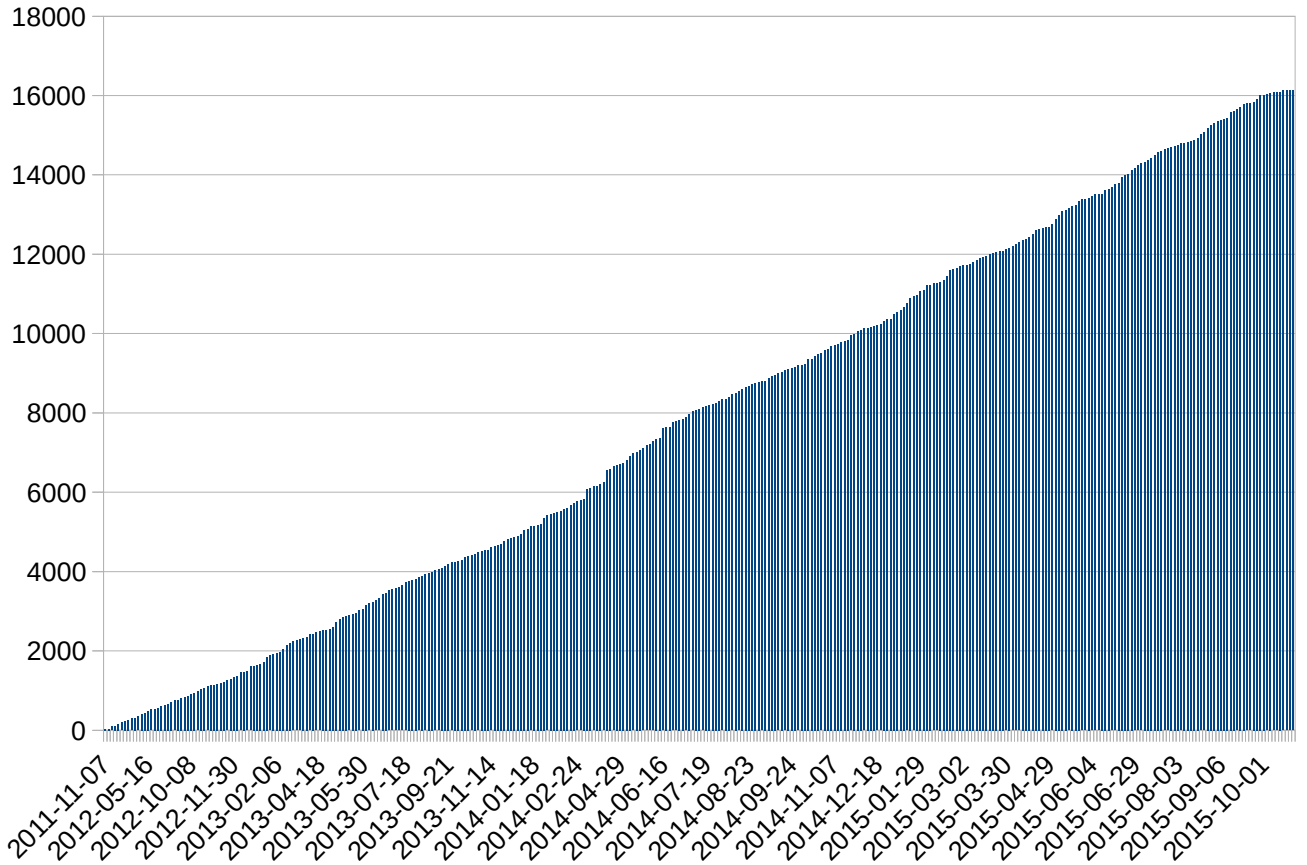

# Workshops

# How It's Going



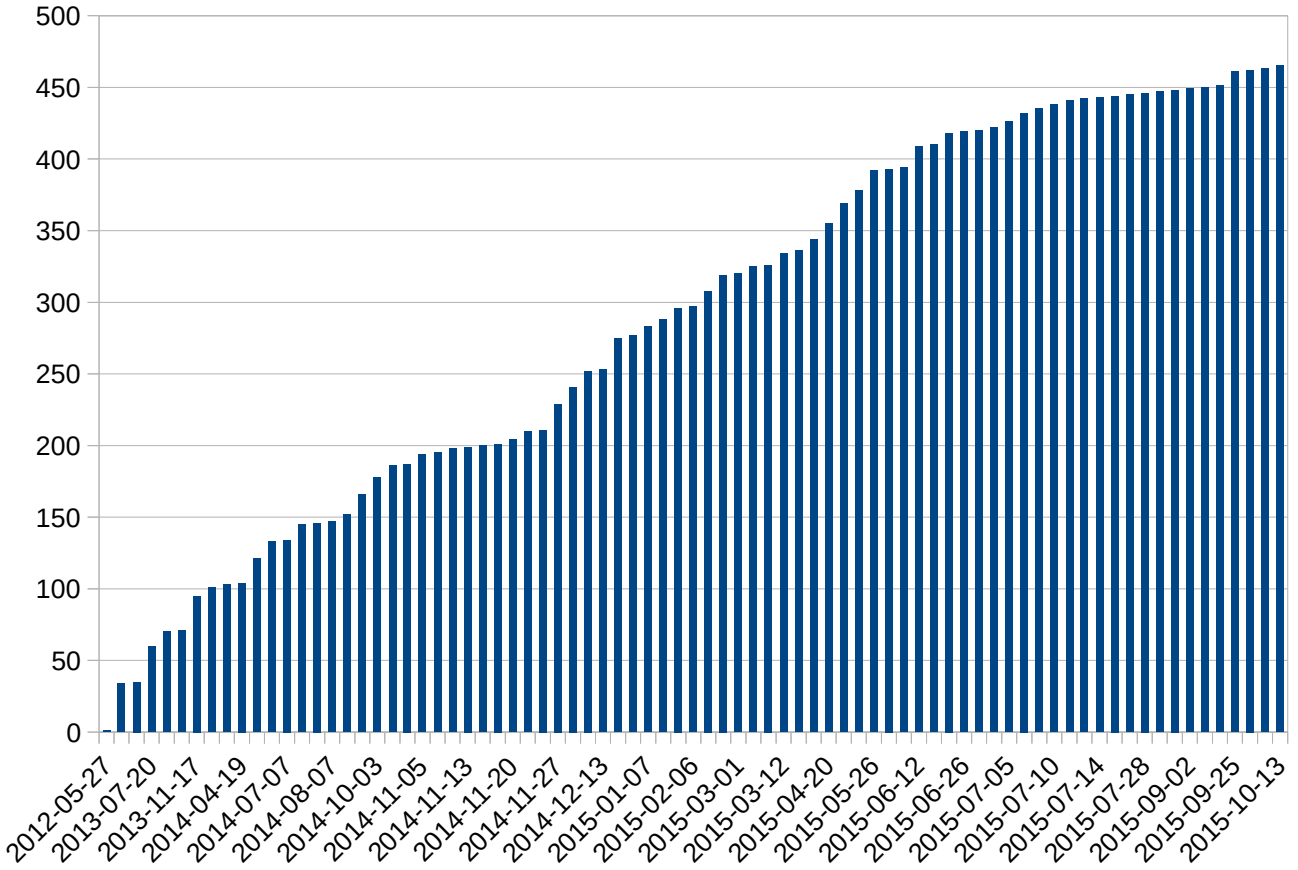

# Workshops

# How It's Going



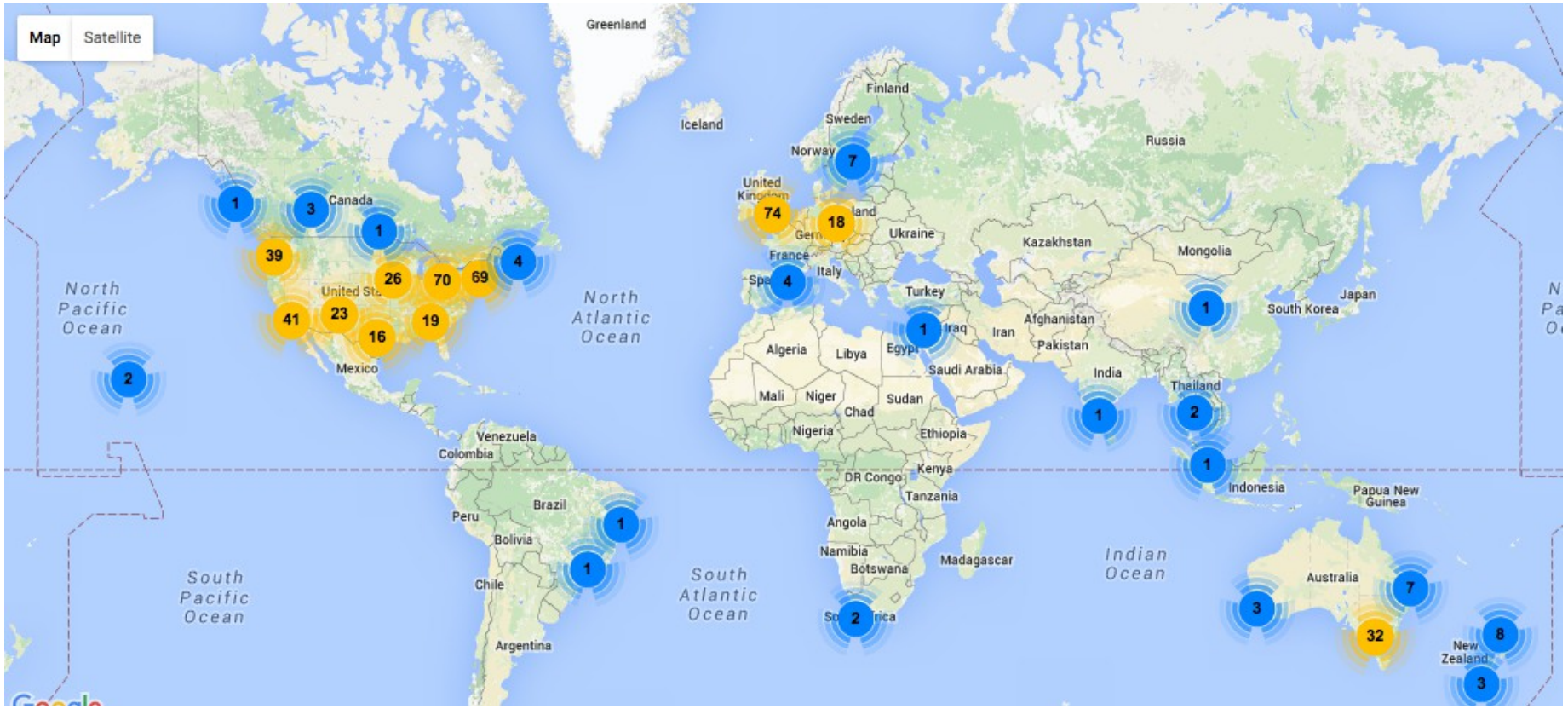

# Learners

# How It's Going




# Instructors

# How It's Going




# Instructors

# What We've Accomplished

- Honestly don't know
- Anecdotally, save people 10-20% of their time for the rest of their careers
- And prepare them for processing petabytes using GPUs in the cloud

# What We've Learned (Version 1)

- Software engineering isn't appropriate for most scientists
- Week-long workshops are easy to schedule, but bad for learning

# What We've Learned (Versions 2-3)

- Hard to fit this into existing curricula
- Hard to convince Computer Science departments to care

# What We've Learned (Version 4)

- Videos aren't cost-effective
- What happens *after* matters as much as what happens *during*

# What We're Learning (Macro)

- Instructor training creates community
- Collaborative lesson development
- Early joiners are atypical
- Every partner has different needs
- Many people would rather argue about technology for a week than spend half an hour fixing a lesson

# What We're Learning (Micro)

- Teach in pairs
- Learners use their own machines
- Live coding
- Sticky notes
- Collaborative note-taking
- Get feedback
- Iterate, iterate, iterate...

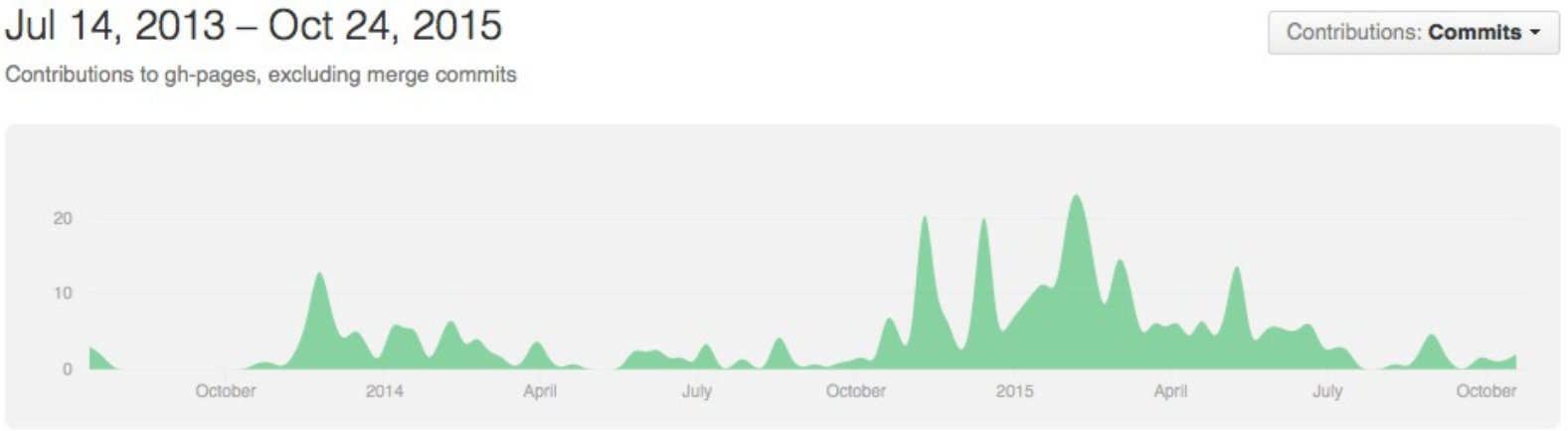
# Why People Volunteer

- Make the world a better place
- Self-defense
- Learn this stuff themselves
- Make new friends
- Boost their careers

# A Puzzle

- Thousands contribute patches to open source software projects
- Millions have edited Wikipedia
- Why don't people build lessons this way?

# It Works




- 187 contributors to our lessons so far this year
- A culture of contribution

# It Spreads


- Domain-specific lessons
- Shared instructor pool
- Next: librarians, humanities, ...

# Why You Should Care

- Extraordinarily cost-effective
- Developing a new model for curriculum development
- Giving *everyone* a say in shaping 21st Century science

# How You Can Help

- Come learn
- Host a workshop
- Become an instructor
- Contribute to our lessons
- Build tools

# Thank You


Aleksandra Pawlik

Piotr Banaszkiewicz

Klemens Noga

http://software-carpentry.org