# Software Carpentry: Lessons Learned

Greg Wilson

software carpentry

5-15%

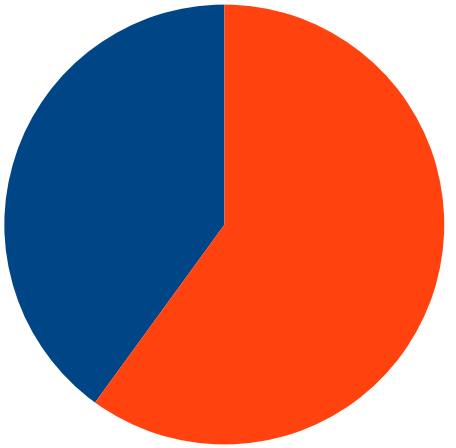
GPU clusters to analyze petabytes in the cloud

85-95%

Sending each other spreadsheets by email

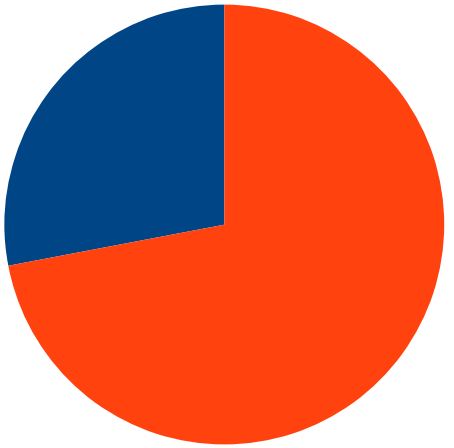
# Surely You're Exaggerating

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



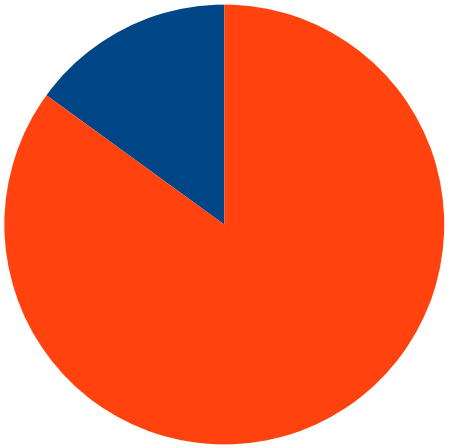
# Surely You're Exaggerating

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



# Surely You're Exaggerating

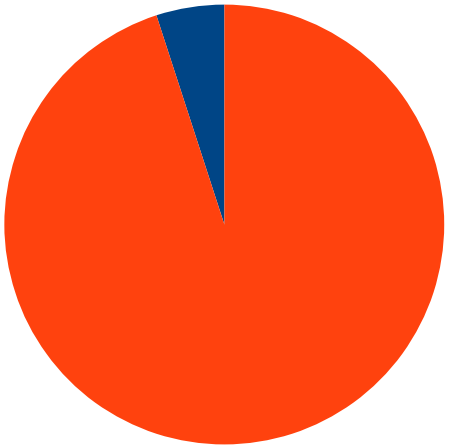
3. How many routinely break large problems into pieces small enough to be
   - comprehensible,
   - testable, and
   - reusable?



# 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?



# 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!

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

*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


software
carpentry

# What We Teach

Unix shell

Git

Python or R

SQL

# What *Actually* We Teach

| | |
|---|---|
| Unix shell | Task automation |
| Git | Tracking / sharing |
| Python or R | Re-use |
| SQL | Data management |

# How It's Going

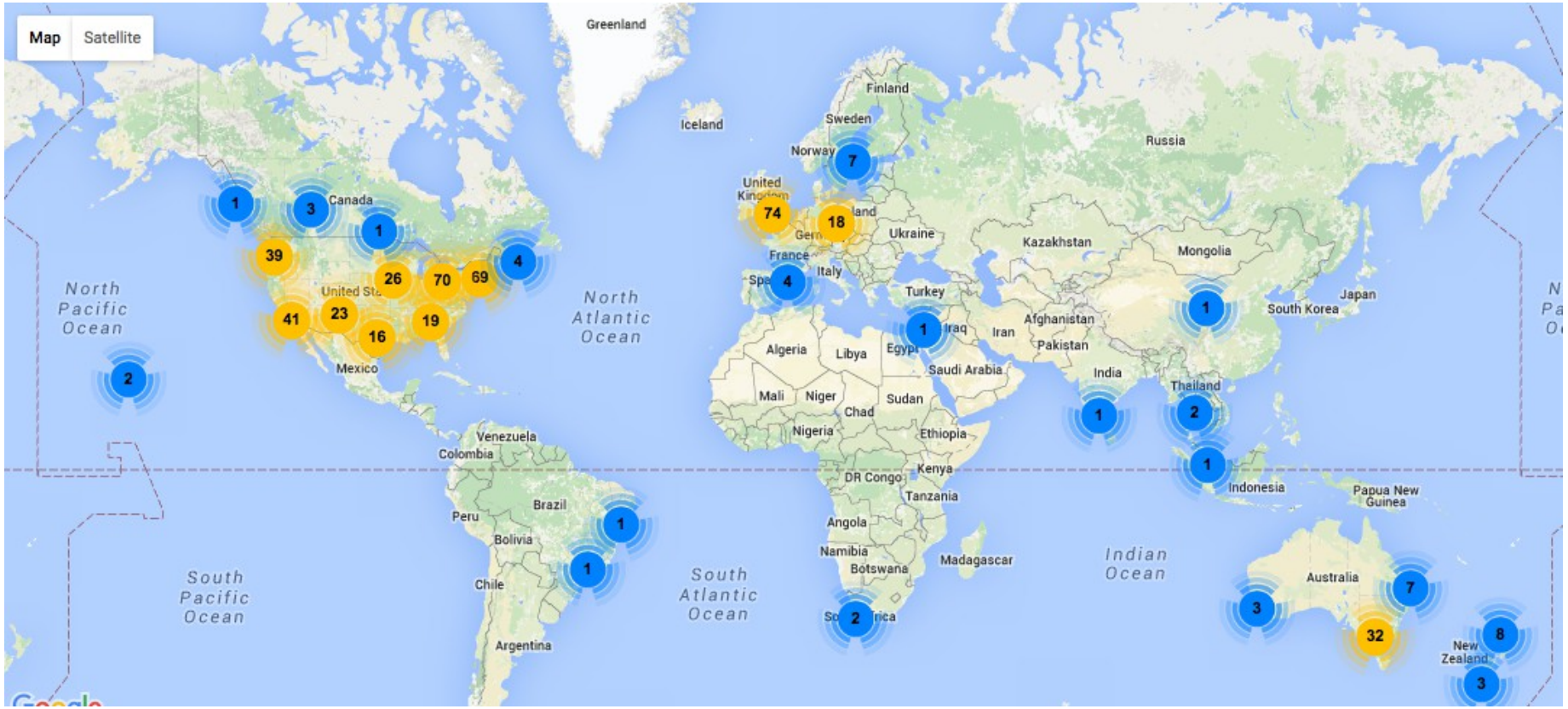
In 2015:

290 workshops

~9500 learners

28 countries

# How It's Going




# Instructors

# What We Learned (Versions 1-4)

- Teach less, learn more
- Videos aren't cost-effective

# And Now... Version 5

- Sticky notes
- Collaborative note taking
- Debriefing

# And Now... Version 5

- Learners should use their own machines
- Never teach alone
- Early joiners are atypical

# And Now... Version 5

- Instructor training creates community
- Live coding is better than slides
- The mistakes *are* the lesson

# And Now... Version 5

- Every partner has different needs
- People would rather argue about technology than pedagogy

# And Now... Version 5

- Iterate
- Iterate
- Iterate

# Why People Volunteer

Make the world a better place

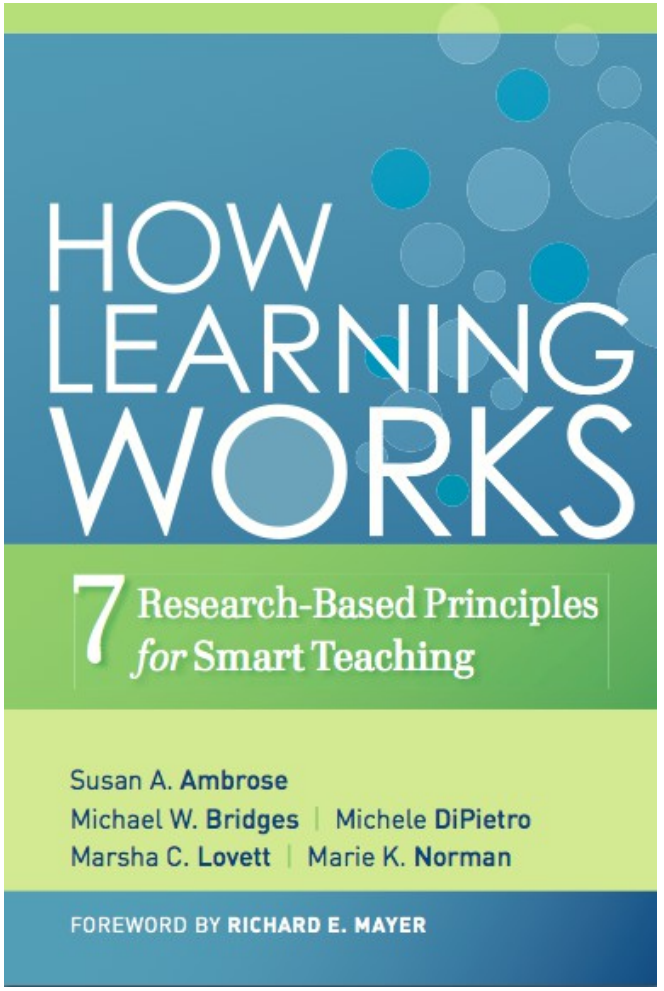
Self-defense

Learn this stuff themselves

Make new friends

Boost their careers

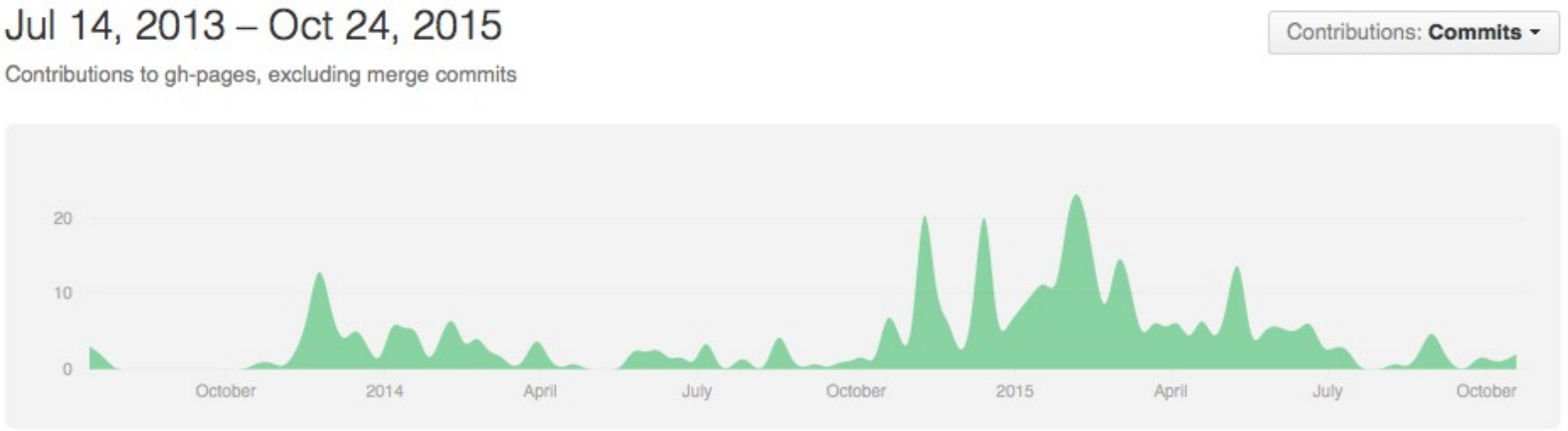
# Learning About Learning



# A Puzzle

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

# All Together Now




- 187 contributors to our lessons in the run-up to publication
- A *culture of contribution*

# The Model Transfers


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

# How You Can Help

Learn

Host

Teach

Write

*Share*

# Thank You



http://software-carpentry.org