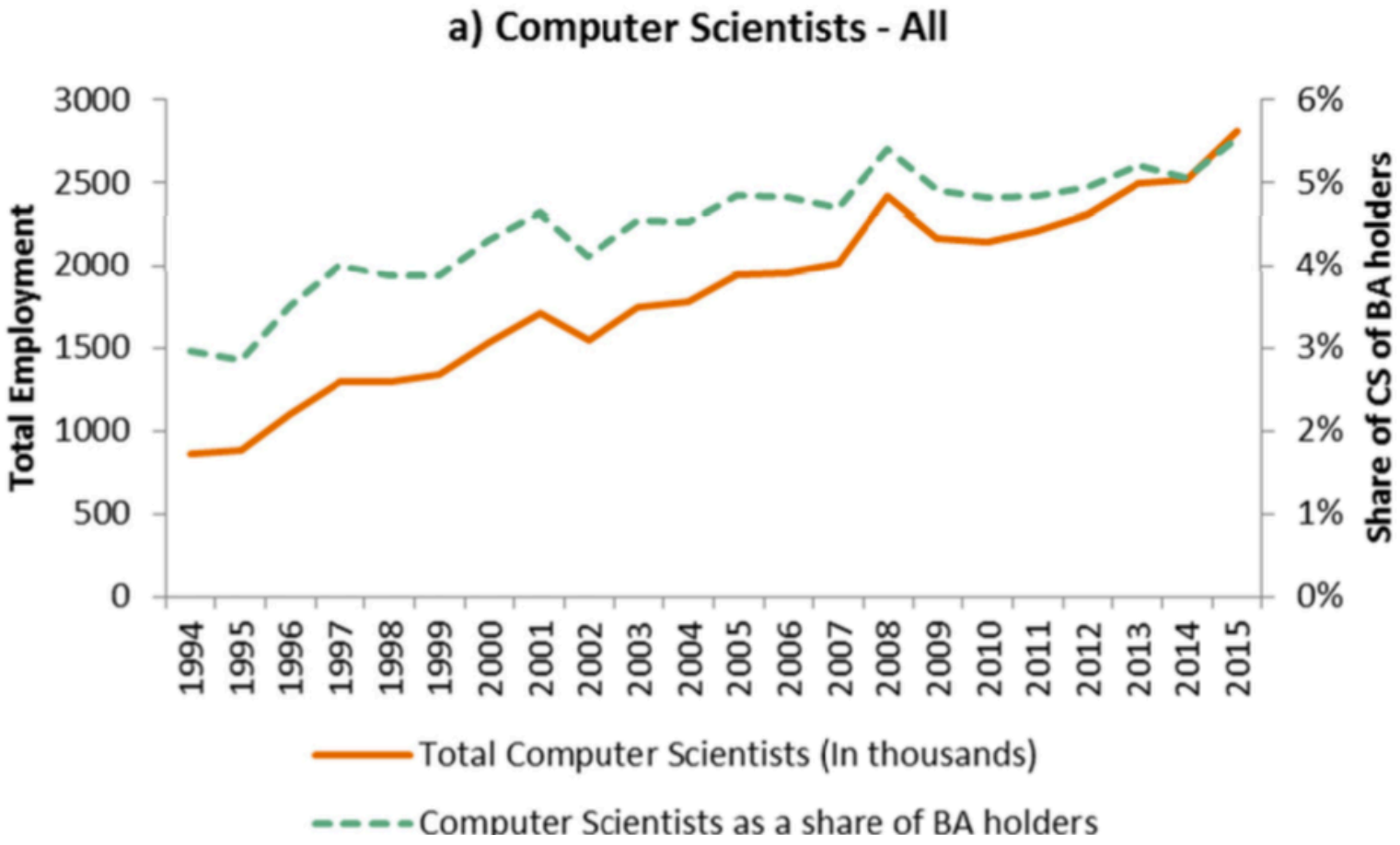
A Case for An Open Source CS Curriculum

Tom Anderson

h/t Aditya Akella, Jeff Chase, Armando Fox, Wyatt Lloyd, Dave Patterson, Geoff Voelker

Is There a Problem?



Bound and Morales, Workforce Trends in Computer Science, 2016
US added an average of ~ 100K new jobs in CS per year since 1994

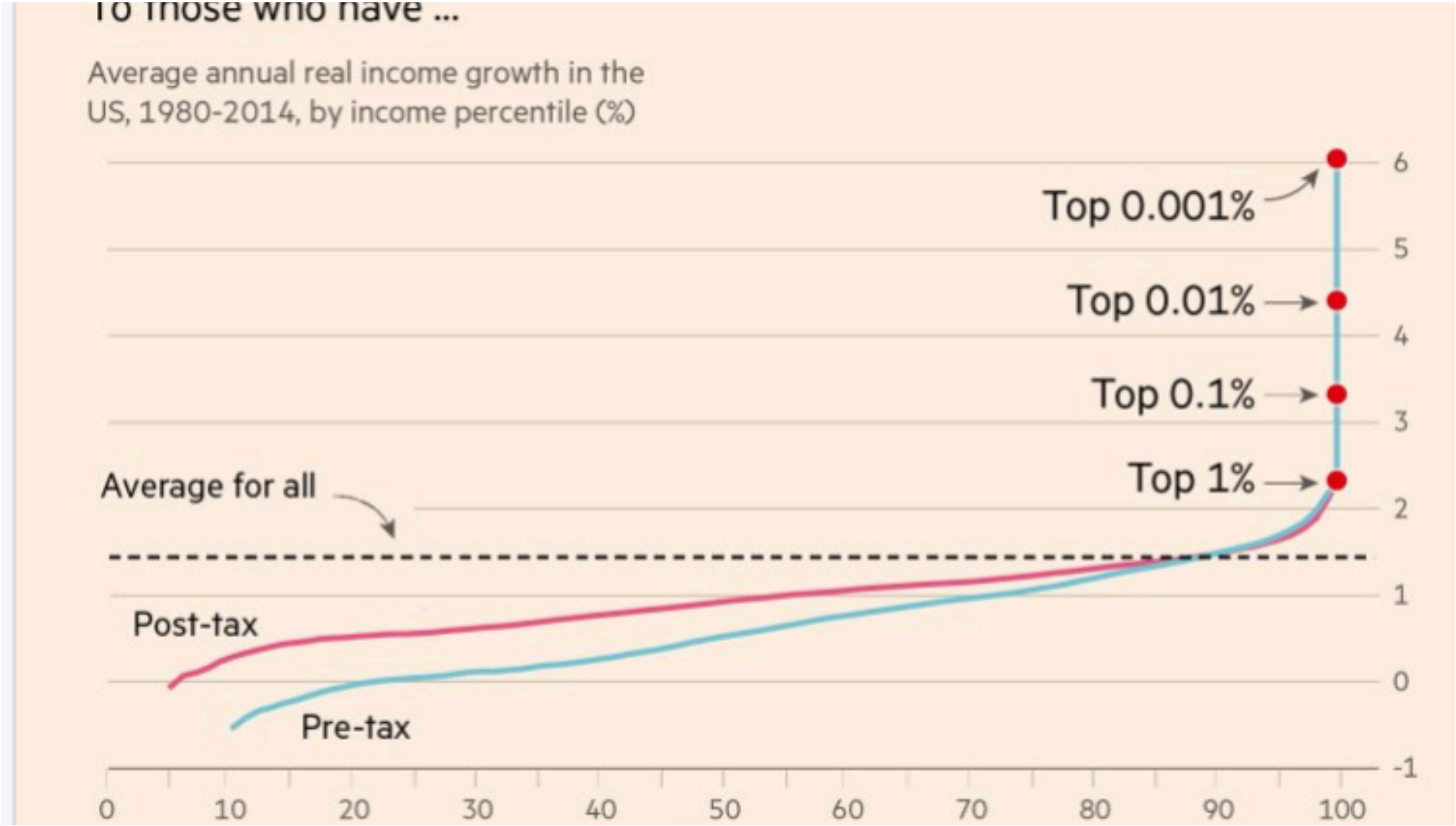
Do We Need That Many CS Grads?

- Supply or Demand?
  - 60K CS bachelors degrees in 2015 in the US
- US added 100K new CS jobs/year (1994-2015)
  - Steady state attrition requires 60K new hires/year
- Estimated 240K CS job openings in US
- Largely not a problem of student interest
  - 15% of entering UW freshmen intend to major in CS
  - Not surprising given the job market for non CS BAs

How Big an Opportunity?

- Impact of supply of tech degrees on metro GDP
  - Peri et al., Journal of Labor Economics, 2015
- 1% increase in tech degrees in metro area
  - 7% increase in avg income, for *other* BA holders
  - 3% increase for *non-college* educated workers
- Similar to estimated benefit to DC of Amazon relo
  - $15B per year by 2030
- Plus benefit to students themselves
  - New Berkeley CS grads 2x salaries of non-CS grads
- Nationwide? 1% of workforce is 1.5M
  - 10 years at 150K degrees/year => $1T/year

The Impact of Technology (1980-2014)



The Heroic Professor?

- Writes papers
- Manages graduate students
- Stays up to date on recent research
- Understands how students learn best
- Determines best way to teach each topic
- Develops great custom-built course projects
- Writes textbooks to share educational knowledge
- Leaps tall buildings

The Reality

- Shortage of faculty for in-demand topics
  - Typical OS teacher took an undergrad OS class
- CS teaching lags technology frontier, badly
  - Almost no one has time for course development
- Assignments and projects aren't widely shared
  - It is a lot of work!
  - And you'll see even more online solution sets
- Even slides are often shared only on Piazza
  - Except among friends, or by textbook authors
- Fewer textbooks are being written

Private Schools Aren't Big Enough

Graduating the most CS/CE BAs (US News top 50, first majors)

1. MIT (31%)
2. Caltech (26%)
3. Stanford (16%)
4. CMU (12%)
5. Princeton (10%)
6. Columbia (10%)

Graduating the least CS/CE BAs (US News top 50, first majors)

18. Chicago (4.6%)
19. Yale (4.5%)
20. Hopkins (4.5%)
21. NYU (4.4%)
22. USC (4.2%)
23. Northwestern (3.7%)

Private schools in top 50: 3600 total BA degrees (2017)

Heroic professor model works for a few students.

Public Universities to the Rescue?

Public colleges produce the large majority of BAs

- 65% of 4 year degrees in the US
- 73% of 4 year engineering and CS degrees

Tier 1 research schools aren't enough

- Top 50 ranked schools in CS (public and private) produced about 15K CS/CE BAs in 2017
- Still far short of workforce need

Need solutions that work inexpensively, at scale, at non-tier 1 schools

Public Research Universities

Increasing technical sophistication of economy

Hire faculty who do both research and teaching

- Develop knowledge that pushes the economy forward
- Train the next generation of students in that knowledge

An education open to all who can benefit from it

- a path to levelling income inequality (in theory)

The Paradox of Public Research Universities

- The public thinks we are being paid to teach
- Students are funding most of the cost of the flagship state universities

- We think of our job as research
- We spend most of our time doing research
- Most of us try to minimize the amount of time we spend teaching

Public Schools

Graduating the most CS/CE BAs (top 30 publics, first majors)

1. Georgia Tech (16%)
2. UCSD (11%)
3. Michigan (8%)
4. UC Irvine (8%)
5. UIUC (7%)
6. Minnesota (6%)

Graduating the least CS/CE BAs (top 30 publics, first majors)

25. Ohio State (3%)
26. Arizona (3%)
27. Washington (3%)
28. Utah (2.9%)
29. UC Santa Barbara (2.7%)
30. UCLA (2.6%)

15% of incoming UW want to major in CS. Most public schools are serving many fewer students than the underlying workforce need.

The CRA Consensus

1. Need more CS BAs
2. Fund CS PhD education via NSF
3. Wait 6 years
4. CS PhDs take faculty jobs
5. They produce more BA degrees
6. Wait 6 years
7. More CS PhDs take faculty jobs

Taulbee Survey

- Faculty at tier 1 research schools graduate an average of 0.3 PhDs/year
  - Fairly consistent across public/private
  - Also across different school rankings
- Only 30% of CS PhDs take academic jobs

$\Rightarrow$Faculty produce an average of 1 new assistant professor every decade or so

Minus faculty leaving for industry

Is NSF the Bottleneck?

- NSF funding for CS research has increased 3.5x since 1994
  - $200M/year in 1994 -> $712M/year in 2017
  - Excluding NSF cyberinfrastructure
- Roughly in line with increase in CS jobs in broader economy
  - 800K in 1994 -> 2.6M in 2015

Recap

- Massive underproduction of CS degrees relative to industry need
  - Large missed opportunity for economic growth
- Tier 1 schools producing too few students
  - Even large increases would only make a dent
- Need solutions that work for mid-tier colleges
  - Typically not where research PhDs go to teach
  - But it is where middle and low income students go to learn

Equality of Opportunity Project

Innovative study of anonymized tax records

- Parent and child tax returns, zip codes
- Anonymized college student records

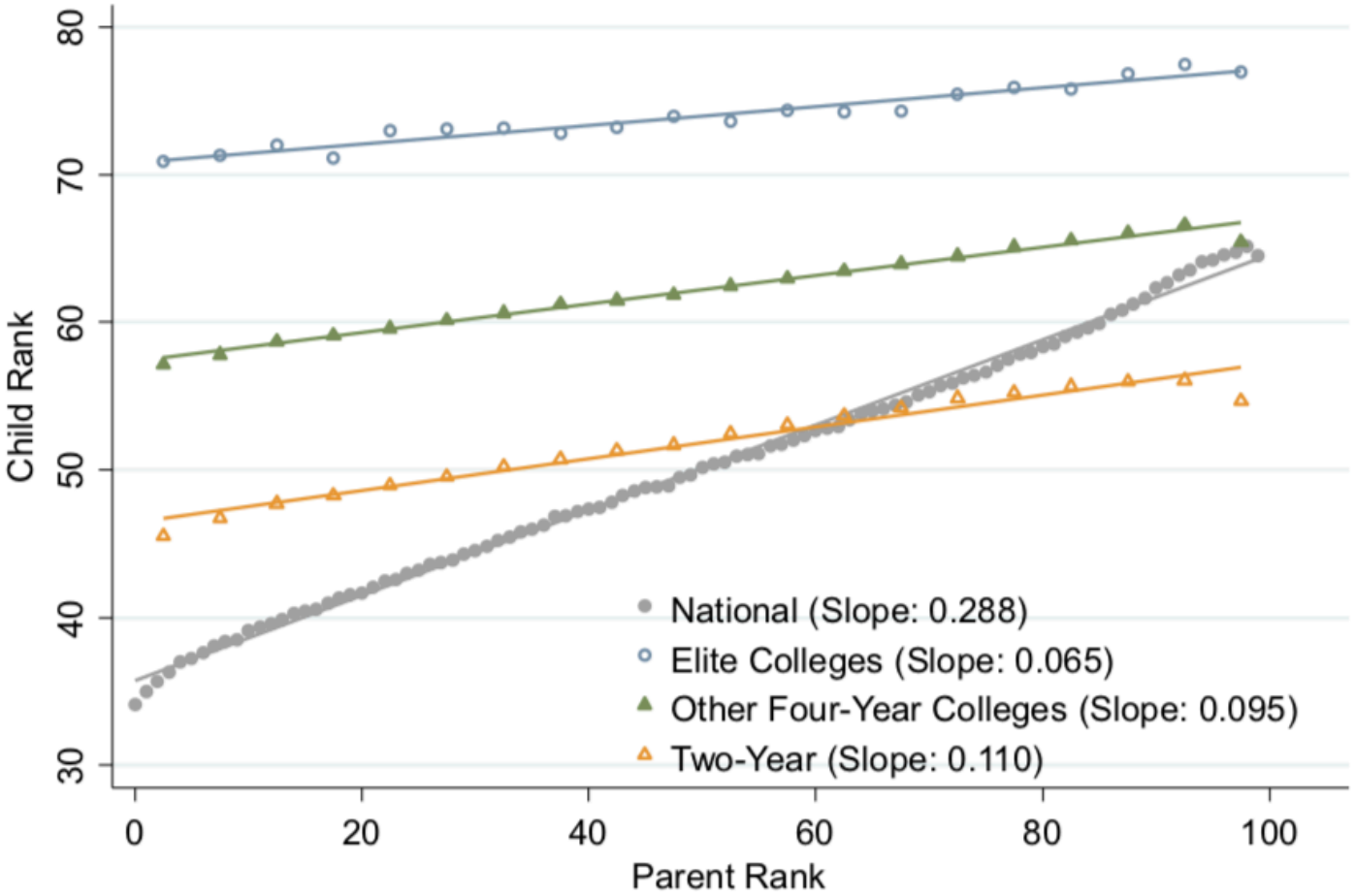
US neighborhoods are segregated by income

- Move a poor child to a wealthy neighborhood
- Outcomes converge at rate of 4%/year

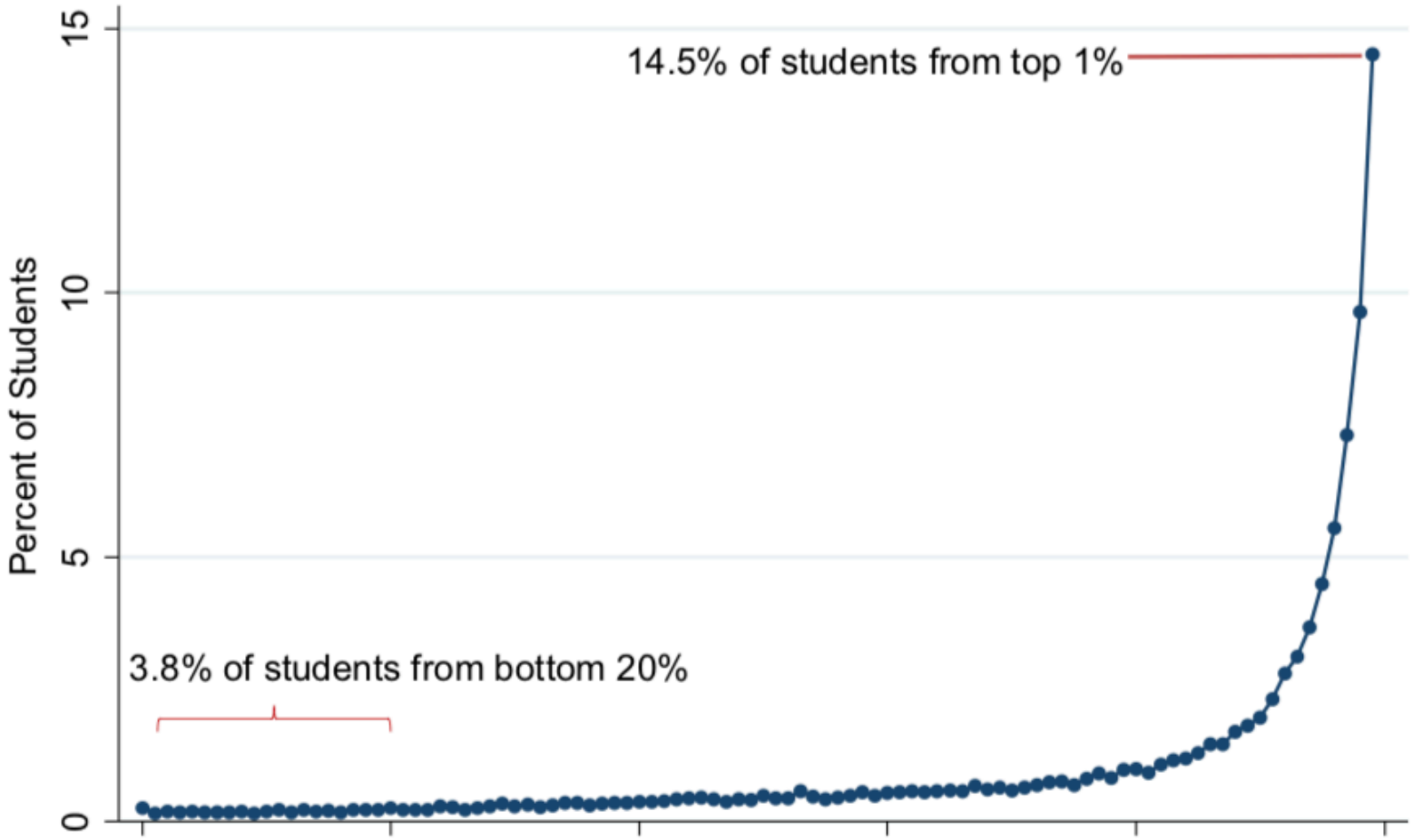
Universities as segregated as neighborhoods

- Few low/middle income students attend tier 1s
- Those who do end up similar to wealthy kids

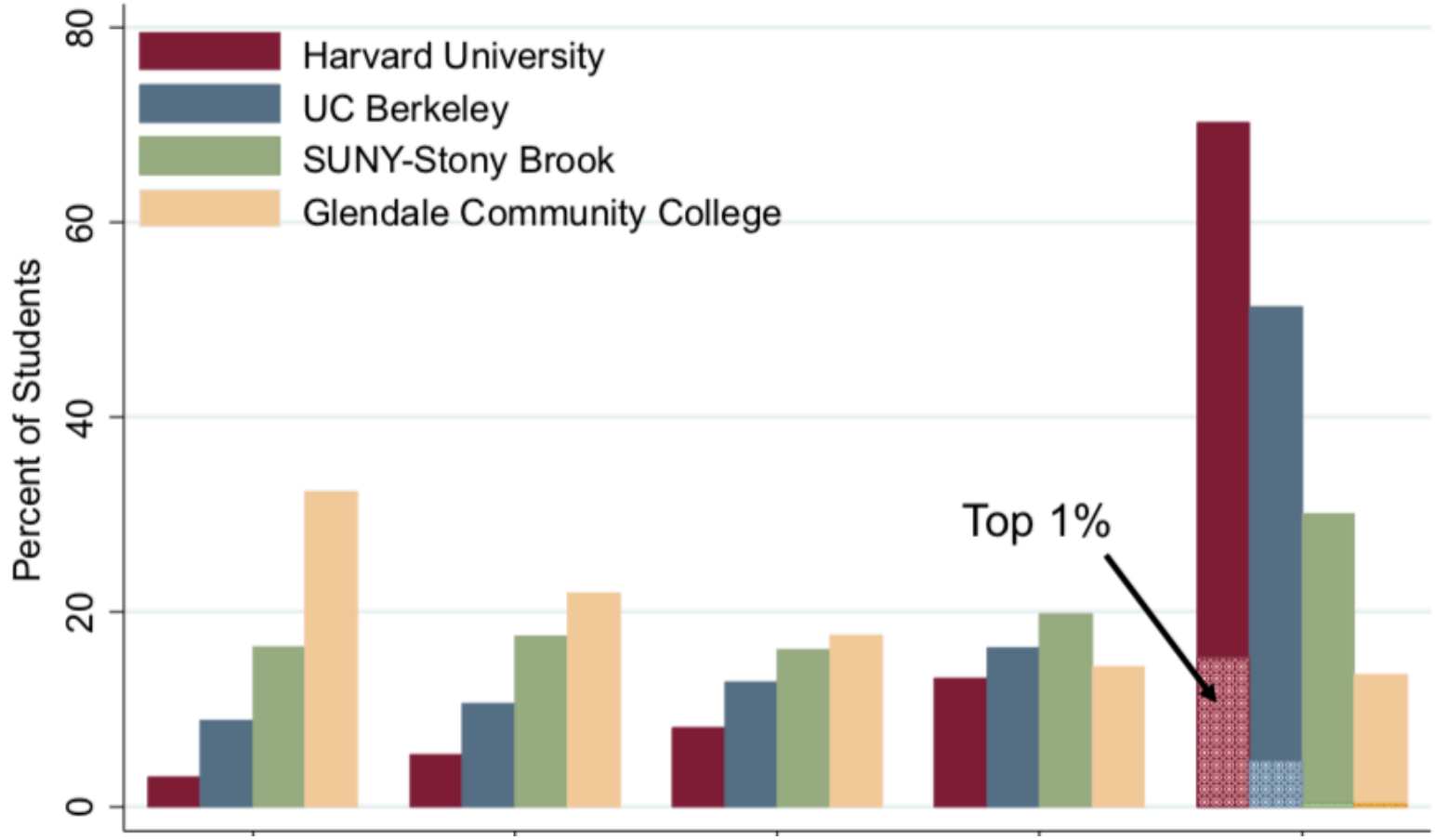
College Can Be A Social Leveller



Distribution of Parent Income, Ivy+



Not Just Ivy+


Harvard University
UC Berkeley
SUNY-Stony Brook
Glendale Community College
Percent of Students
0
20
40
60
80
Top 1%

Top CS Research Universities
US students from top 10% / bottom 60%

Public		**Private**	
UW	1.2	Stanford	2.8
Berkeley	1.3	MIT	1.8
UCLA	1.0	Princeton	4.2
UCSD	0.6	CMU	2.6
Michigan	2.9	Cornell	2.5
Texas	1.4	Caltech	4.5
GaTech	1.8		

Data for students born in 1991, includes (most) transfer students
International students at public schools are likely even more skewed

Top CS Research Universities
Median Family Income (US only)

Public		**Private**	
UW	$113K	Stanford	$168K
Berkeley	$120K	MIT	$137K
UCLA	$105K	Princeton	$186K
UCSD	$ 82K	CMU	$155K
Michigan	$154K	Cornell	$152K
Texas	$124K	Caltech	$146K
GaTech	$130K		

Data for students born in 1991, includes (most) transfer students
International students at public schools are likely even more skewed

Top Research Universities
Pell Grants, all undergrads

Pell grant threshold is roughly median income (due to ARRA)

Public		**Private**	
UCSD	34%	MIT	17%
UCLA	34%	Cornell	16%
Berkeley	28%	Princeton	16%
Texas	24%	Stanford	15%
UIUC	21%	Caltech	12%
UW	21%	CMU	12%
GaTech	15%		
Michigan	15%		
Wisconsin	13%		

College Choices of Low Income Students (Hoxby and Avery 2013)

- 80% of highly qualified low income students are undermatched in school quality
  - Top 4% of SAT scores, high GPA in high school
  - That is, very strong students
- Most low-income students apply only to colleges within 50 miles of home
  - Going away to college is often not an option
- Need to teach students where they live

A Few Examples

- University of Central Florida
  - 38% Pell grant; 37% URM
  - Graduated 480 CS+CE majors in 2017 (3.6%)
- Texas A&M Commerce
  - 47% Pell; 43% URM
  - Graduated 42 CS majors (2.5%)
- UMass Boston
  - 40% Pell; 31% URM
  - Graduated 37 CS majors (1.4%)

Fallacy of Elite Projection

Belief that public services should be designed to be attractive to the elite

- World class airports, train stations, universities, …
- Means fewer services get built
- Leaves most citizens behind

Instead, we need solutions that are designed to work for everyone

Geographic Dispersion

- Software is eating the world
  - Becoming an essential element of every major enterprise
  - Ex: manufacturing, agriculture, health care management, elections…
- Geographically dispersed economic benefits if we can create a dispersed knowledge base
  - Not everyone will move to the valley

Types of Teaching Knowledge (h/t: CMU Eberly Center)

- How to teach in general
  - Not my expertise!
- Subject knowledge
  - Concentrated at tier 1 schools
  - Which are highly income segregated
- How to teach subject (in context)
  - Also concentrated at tier 1's, but lessons may not apply to all audiences

Coursera/EdX?

- Successful model for students who are already self-guided learners
  - Wherever they live around the globe
- What about everyone else?
  - Need a human being to help problem solve
  - Why are my students having difficulty with a concept? How can I explain it better to them?
- Not a curriculum
  - Most of the effort at CS1/CS2 and specific skills, partly due to the economic model

Flipped Classrooms?

- Using someone else's videos at another institution?
  - Undercuts the authority of the teacher
  - Still need local problem solving and adaptation
  - May work well for CS1 outreach to high schools
- Using someone else's textbook/assignments/ autograding scripts at another institution?
  - This often works really well
  - Fewer bugs, clearer expectations, …

Tailored Instruction

- Controlled experiment at Texas-Austin (Psych 1)
  - Lecture vs. lecture + online quiz about previous lecture
  - Quiz auto-tailored to the student's comprehension, based on previous answers
  - Almost no effect on grades for high income students
  - About a half a grade point difference in GPA for low income students => impact on other classes!
- In the skill set of every instructor
  - Outside of the time budget of almost every instructor

SIGCSE

- Isn’t there already a conference dedicated to educational issues?
  - Almost exclusively attended by instructors
  - Very little mixing with research faculty
- Lots of innovation for CS 1, CS 2
  - Many competing alternatives
- Not as much progress as you go higher into the curriculum
  - Not for a lack of trying

What is to be done?

- Solutions that are effective at scale
- At schools without research programs
- With the faculty at those schools as allies
- Without spending a lot of money
- And you won't get rich or famous

- Increase number of CS students nationwide
- Improve quality of CS teaching nationwide
- Improve income diversity of CS graduates
- Improve ethnic diversity of CS graduates

What is to be done?

- How do we help ranked schools teach more students and more advanced material?
  - Many already teaching very large classes
  - Lots of time spent treading water
- How do we help mid-tier schools teach more students and more advanced material?
  - Often teach multiple classes/term
  - Lots of time spent treading water

We Have a Lot Already

- For some topics, textbook authors have already put together most of what we need
  - Examples: CS:APP, others
- For some topics, projects in wide use
  - Some even with autograding
- For other topics, there's work to do
  - Especially true in systems, imo
  - It's a large step from local use to global use
- What if we coordinated our efforts?

An Open Source CS Curriculum

- An entire undergraduate curriculum
  - Roughly 20 courses
- Focus on course software, not lecture delivery
  - Every school has teachers
  - Need to enable them, not compete with them
  - Need range of assignment difficulty
- Course in a box
  - Problem sets, programming modules, documentation
  - Automated grading, autotuning of difficulty
- Open source: design for change

Open Source: Linux Model

Architect for extensibility

- Ex: adding a new file system, or a new packet queue scheduler, or a new device driver
- Allows large number of people to contribute

Core architecture team

- defines APIs
- decides which community contributions are adopted into mainline source tree

Anyone can fork and customize

- Ethic of adoption of community contributions

A Curriculum

- Hard to make course materials effective in isolation
  - What have students already mastered?
- Need the set to hang together as a group
  - An answer to: what do I need to do to learn CS?
  - Requires coordination
- Enable others to contribute content
  - Ex: MIT has a number of P/F skills courses offered by students for students
  - Loosen idea that only faculty set standards

How Many Courses?

At UW, 20 (quarter) courses taken by a majority of majors

- CS 1
- CS 2
- Web programming
- Discrete math
- Probabilistic reasoning
- Data structures
- Software design and testing
- Machine structures
- Systems programming
- Programming languages
- Database systems
- Algorithms
- Database implementation
- Operating systems
- Distributed systems
- Computer security
- Networks
- AI
- Machine learning
- Computer vision

Case Study: Distributed Systems

Grew course from 42 to 175 in 3 years

- TA's entirely self-generated
- Build a dynamically sharded linearizable and highly available key-value store with multikey transactions

Oddity distributed systems debugger

Model checked, autograded assignments

- Students self-diagnose

Lectures and labs online

- And solution sets ☹

Community Colleges

- Can we extend this model to non-4 year degrees?
  - Two year program in math, data science, and computing
  - With employers at the other end
- Improve graduation rates
  - Focus teachers on helping students
  - Not on grading, assignment creation, …
- Continuing ed is part of their charter
  - Not just for young adults
- Berkeley is prototyping this
  - ¼ of all community college students are in CA

What About Copying/Cheating?

- We're losing the battle
  - Posting solutions, and using posted solutions, is endemic
- Students have an exit test: their job interview
  - Enhanced by having a standard curriculum!
  - Copying often triggered by lack of help and unclear expectations
- Reduce salience of project grades?
  - Exams can test whether student mastered the material
- Centralize tools for copy detection
  - Teachers don't have enough time to make this a priority

Cost

- Developing a 2-week assignment, hosted in the cloud with documentation and autograding
  - A small integer number of person-months of effort
  - Each course needs four or five of these, possibly building on top of each other
- Plus common software infrastructure
- About the cost of an NSF Expedition, in total across all subjects

Next Steps

- Community building
- White paper explaining the effort
- Workshop to organize a proposal
- Contact me if you are interested

Summary

We have an opportunity for a large step improvement in computer science education on a national scale, by taking the effort we are already putting into course software, and organizing it a bit differently