COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

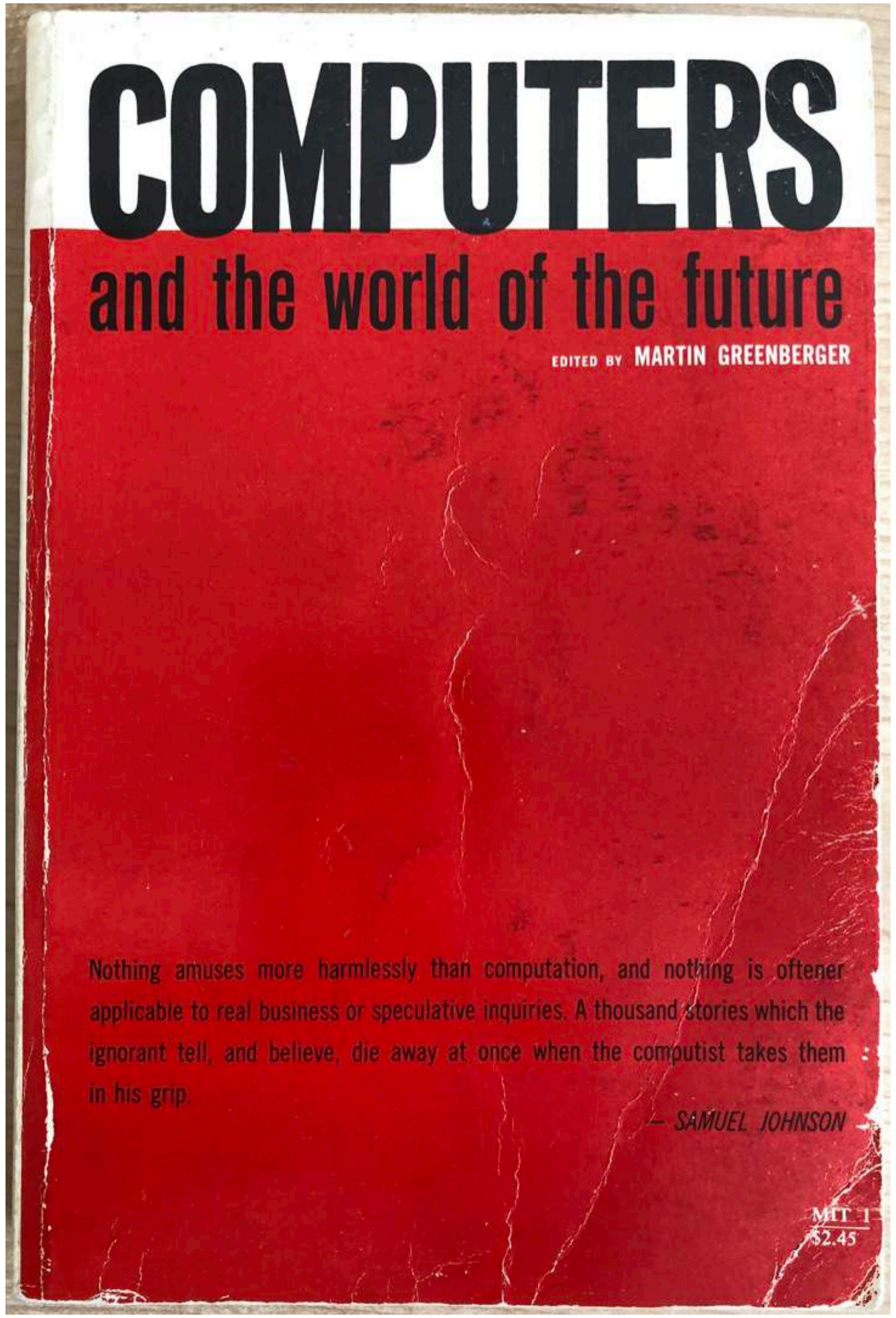
Adding a Teaspoon of Computing to History and Mathematics Classes

Mark Guzdial

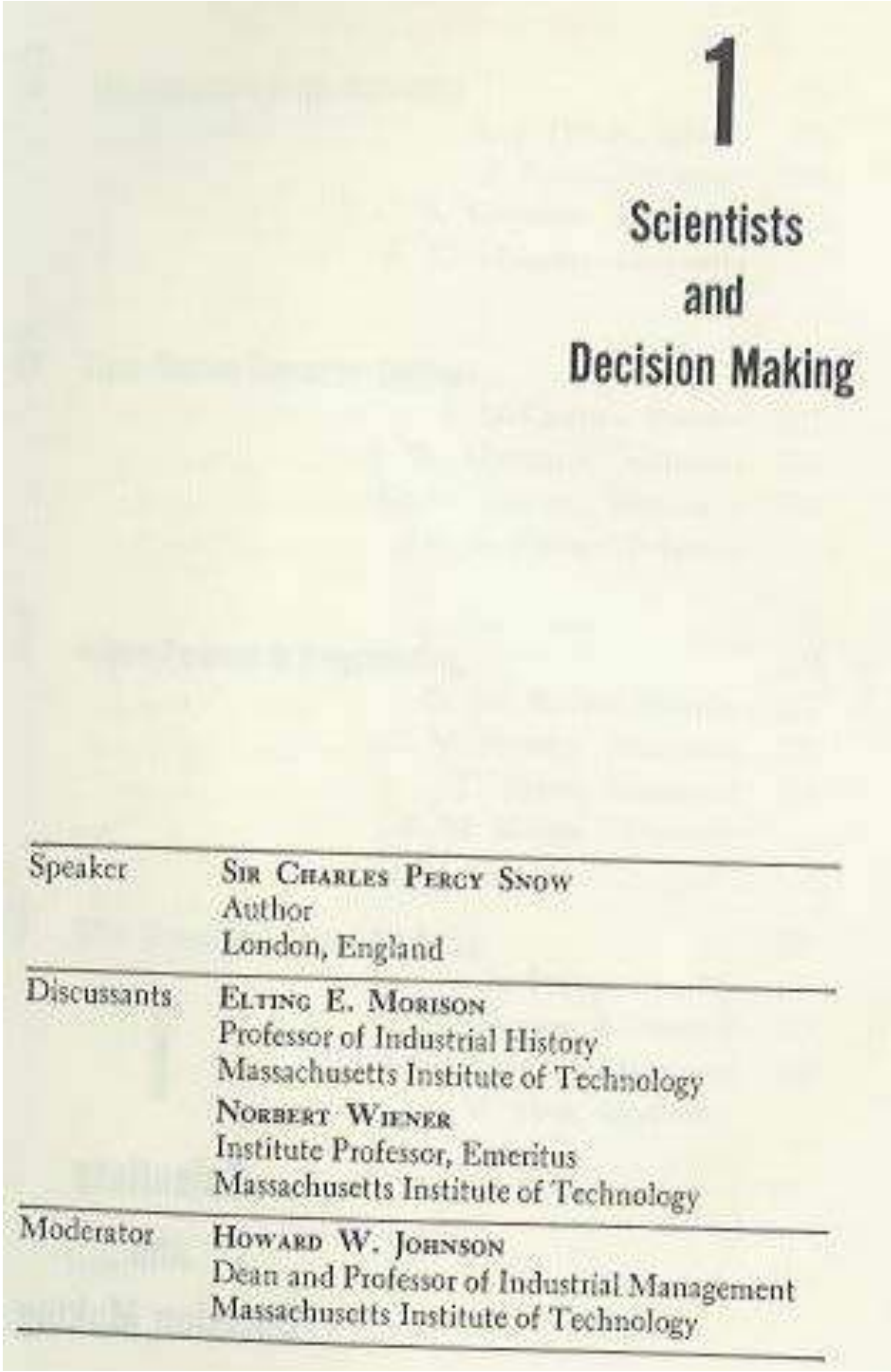
1959

Today's Story

- Computing was created to be taught to *everyone*.
- Are we reaching *everyone* now? (Hint: "No.")
- Teaspoon languages as a way to change computing to reach everyone.
  - For History: DV4L
  - For Mathematics/Engineering: Pixel Equations
  - For Mathematics/Combinatorics: Counting Sheet
- Big question: What are students learning?



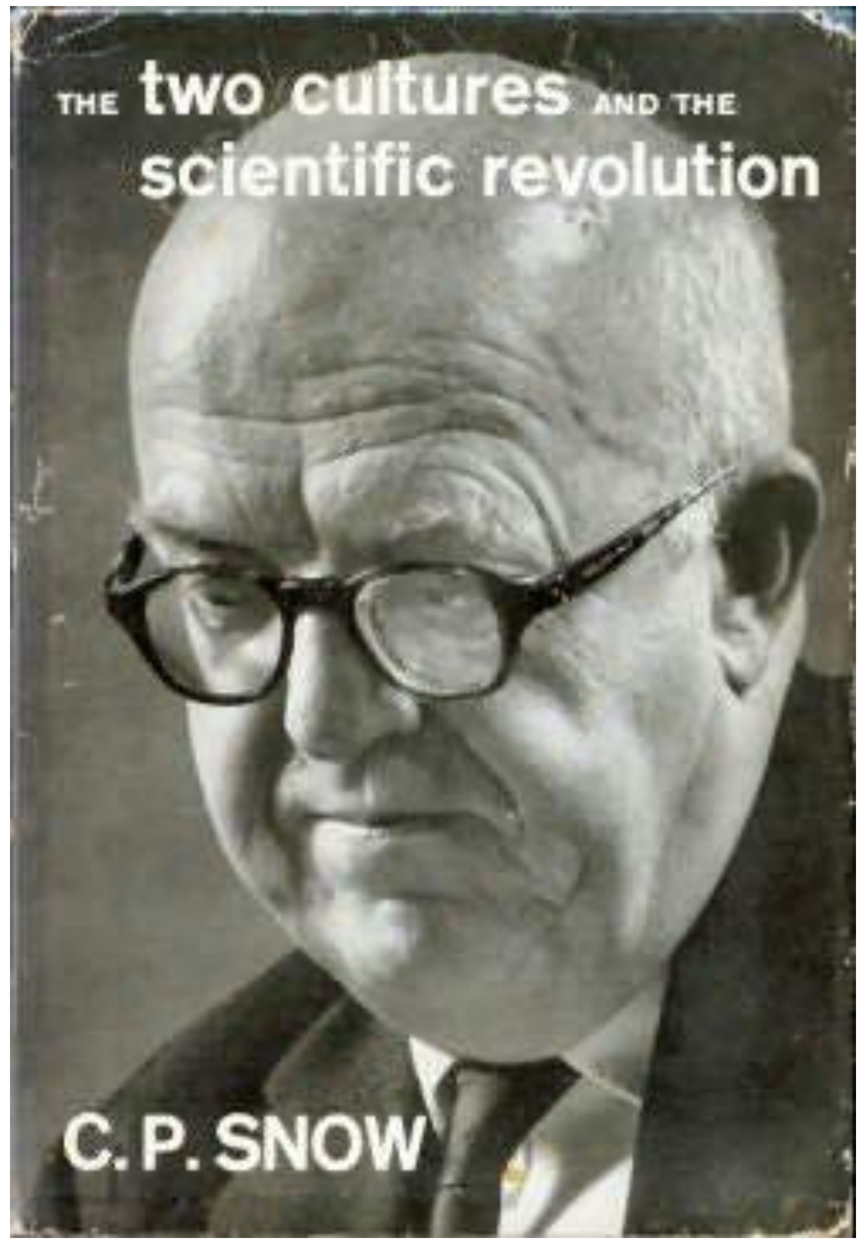
1961



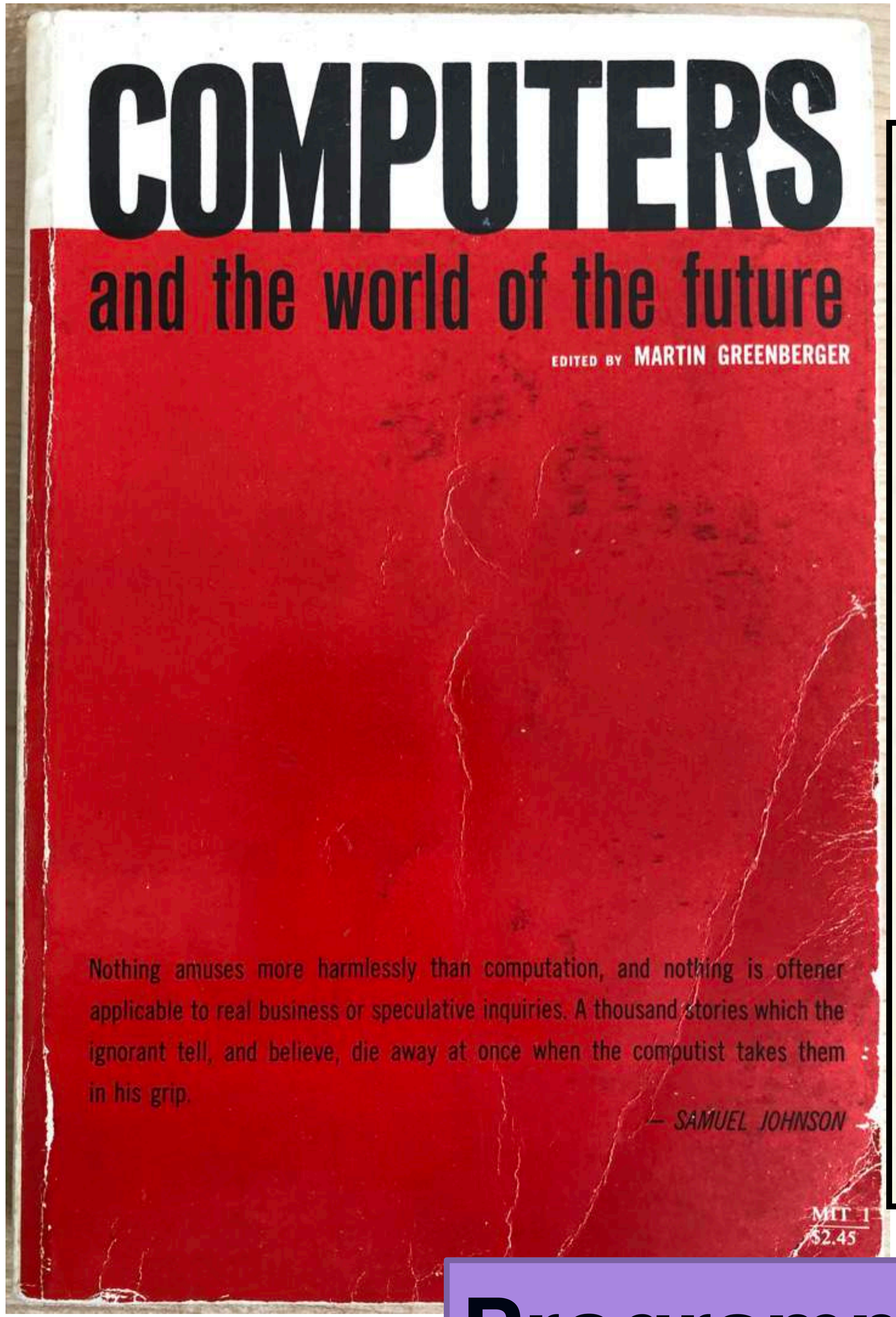
1

Scientists and Decision Making

Speaker	Sir Charles Percy Snow Author London, England
Discussants	Elting E. Morison Professor of Industrial History Massachusetts Institute of Technology Norbert Wiener Institute Professor, Emeritus Massachusetts Institute of Technology
Moderator	Howard W. Johnson Dean and Professor of Industrial Management Massachusetts Institute of Technology



“A handful of people, having no relation to the will of society, having no communication with the rest of society, will be taking decisions in secret which are going to affect our lives in the deepest sense."



1961

5

The Computer in the University

Speaker	ALAN J. PERLIS Director of the Computation Center Carnegie Institute of Technology
Discussants	PETER ELIAS Head, Department of Electrical Engineering Professor of Electrical Engineering Massachusetts Institute of Technology J. C. R. LICKLIDER Vice President Bolt Beranek & Newman Inc.
Moderator	DONALD G. MARQUIS Professor of Industrial Management Massachusetts Institute of Technology

Alan Perlis

Programming changes how we understand

Photo: CMU

First published definition of Computer Science

"The study of computers and all the phenomena surrounding them."

***Science*, 1967**

Alan Perlis

Herb Simon

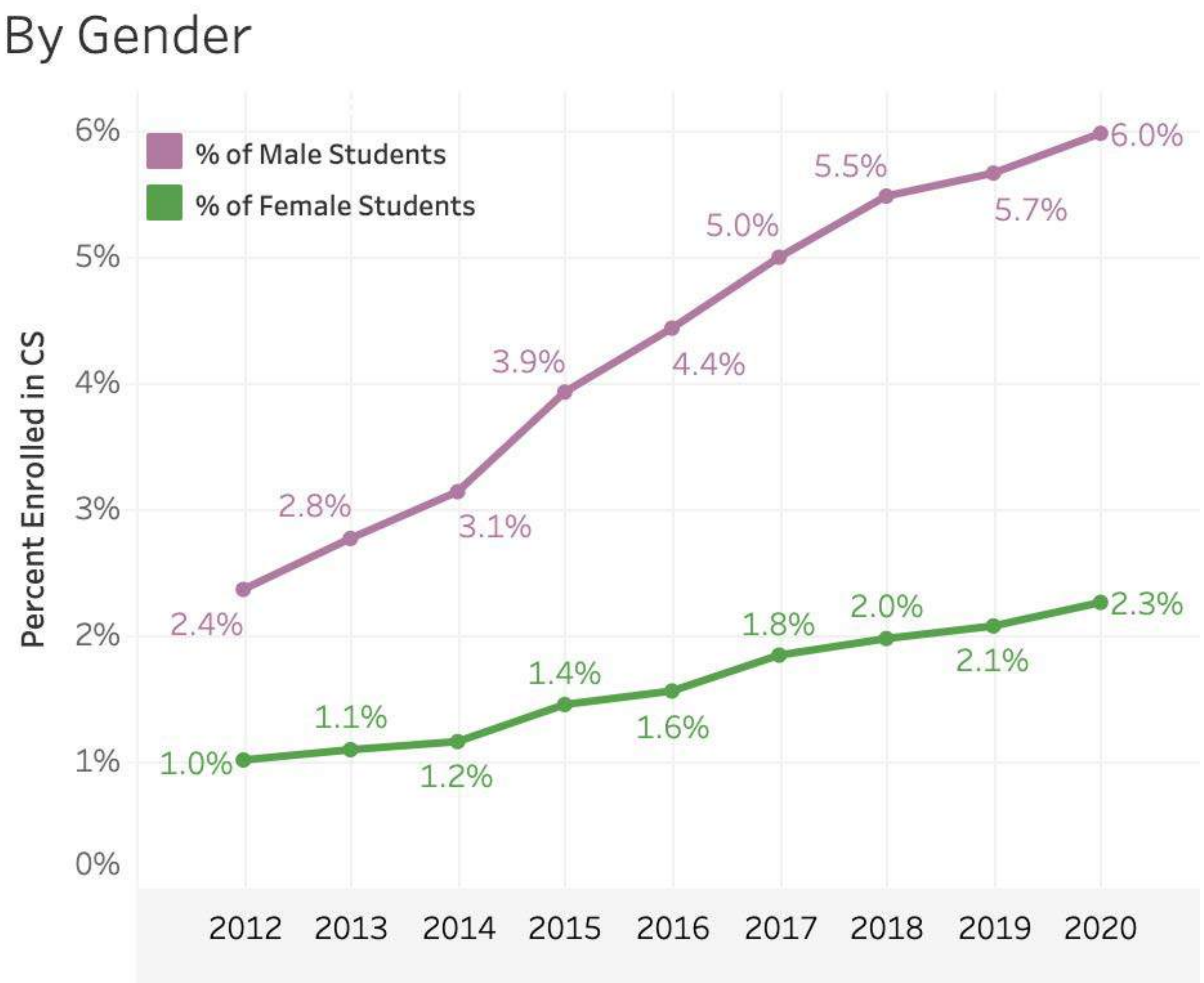
Alan Newell

This is broader than how most people define computer science today.
Let's call this *Computing*

4.7%

Percentage of US high school students enrolled in a CS course

4.7%



By Gender
% of Male Students
% of Female Students
Percent Enrolled in CS
6%
5%
4%
3%
2%
1%
0%
2012
2013
2014
2015
2016
2017
2018
2019
2020
2.4%
2.8%
3.1%
3.9%
4.4%
5.0%
5.5%
5.7%
6.0%
1.0%
1.1%
1.2%
1.4%
1.6%
1.8%
2.0%
2.1%
2.3%

Computer science in high schools is growing very slowly

- In England (from Roehampton Report 2018):
  - 53% of schools offer CS GCSE, 12% of students take it.
    - < 20% female
  - 36% offer A Level CS, under 3% take it.
    - < 10% female

Data from Peter Kemp

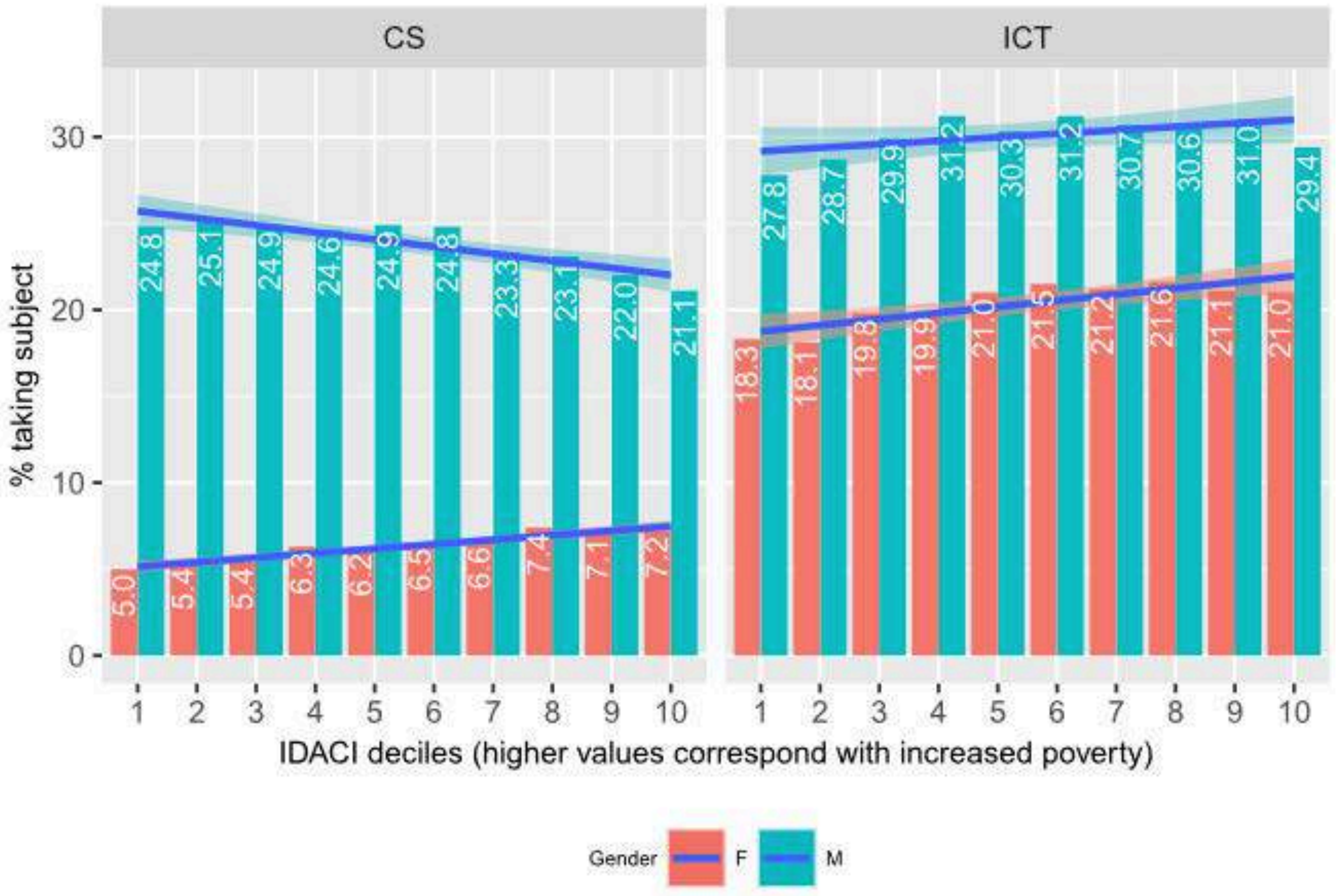


Fig. 2. GCSE computer science and ICT: influence of IDACI on uptake by gender.

BOTTOMLINE:

THE MAJORITY OF SECONDARY SCHOOL STUDENTS IN THE US AND ENGLAND HAVE NEVER SEEN COMPUTER SCIENCE

AP US History vs. AP CS Principles

399K vs 114K

>50% female vs. <30% female

6x more Black

14x more Hispanic

Teaspoon Languages

- A Teaspoon language is a *task-specific programming (TSP)* language — specification of process to be executed by a computational agent.

Adding a teaspoon of computing to other subjects.

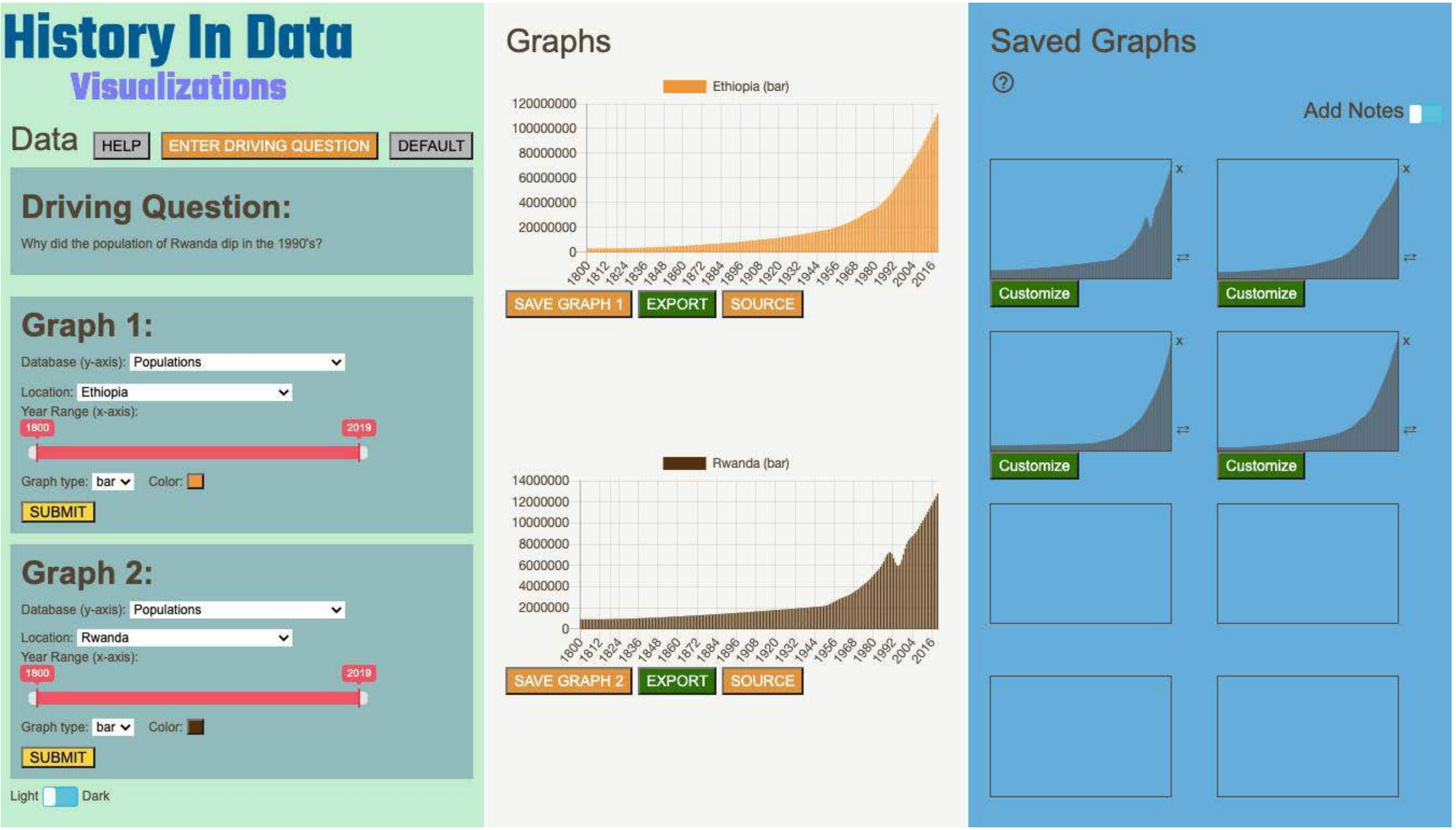
- **USEFUL:** Supports a task (learning activity) that an other-than-CS teacher wants to achieve.

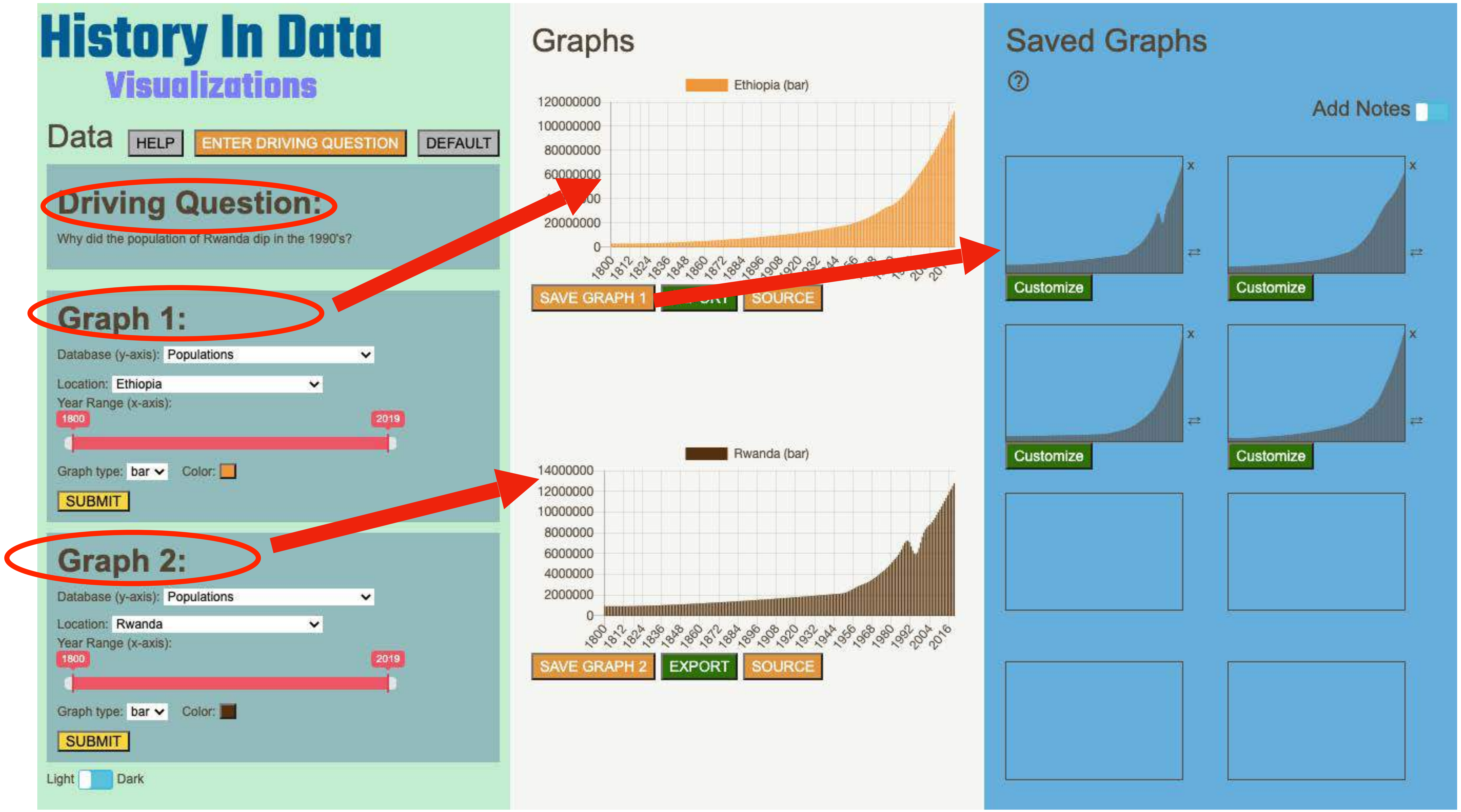
- **USABLE**: Can be learned in less than 10 minutes

#1: DV4L: Data Visualization for Learning

For History Courses

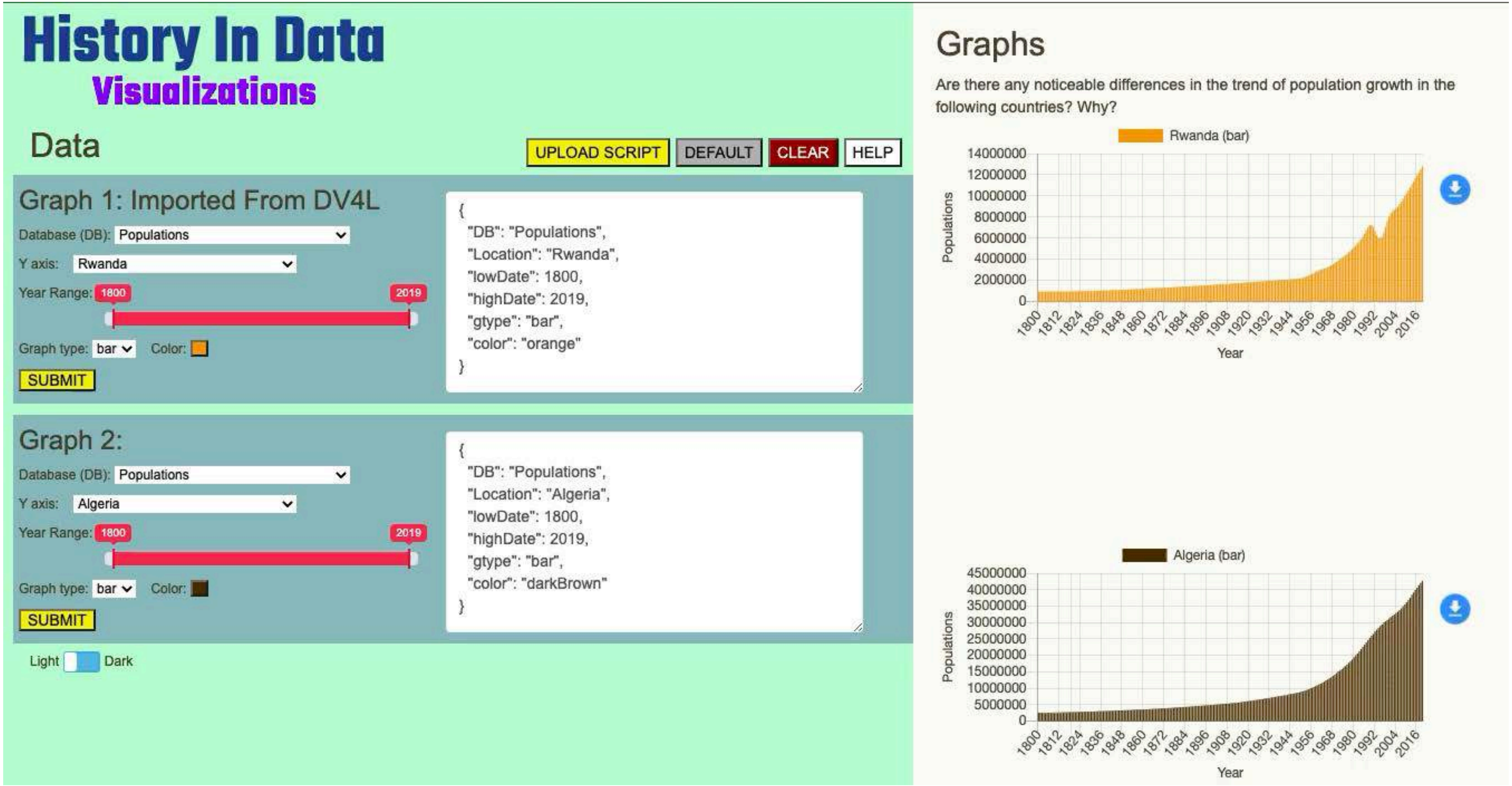
Collaboration with Tammy Shreiner





History In Data
Visualizations
Data
HELP
ENTER DRIVING QUESTION
DEFAULT
Driving Question:
Why did the population of Rwanda dip in the 1990's?
Graph 1:
Database (y-axis): Populations
Location: Ethiopia
Year Range (x-axis):
1800
2019
Graph type: bar
Color:
SUBMIT
Graph 2:
Database (y-axis): Populations
Location: Rwanda
Year Range (x-axis):
1800
2019
Graph type: bar
Color:
SUBMIT
Light
Dark
Graphs
Ethiopia (bar)
SAVE GRAPH 1
SOURCE
Rwanda (bar)
SAVE GRAPH 2
EXPORT
SOURCE
Saved Graphs
Add Notes
Customize
Customize
Customize
Customize

1968
1982
1996
2010
{
"DB": "Populations",
"Yaxis": "Rwanda",
"lowDate": 1800,
"highDate": 2019,
"gtype": "bar"
}
X
Customize
Modify the json code in the scripting version of DV4L


History In Data
Visualizations
Data
UPLOAD SCRIPT
DEFAULT
CLEAR
HELP
Graph 1: Imported From DV4L
Database (DB): Populations
Y axis: Rwanda
Year Range: 1800 2019
Graph type: bar Color:
SUBMIT
{
"DB": "Populations",
"Location": "Rwanda",
"lowDate": 1800,
"highDate": 2019,
"gtype": "bar",
"color": "orange"
}
Graph 2:
Database (DB): Populations
Y axis: Algeria
Year Range: 1800 2019
Graph type: bar Color:
SUBMIT
{
"DB": "Populations",
"Location": "Algeria",
"lowDate": 1800,
"highDate": 2019,
"gtype": "bar",
"color": "darkBrown"
}
Light Dark
Graphs
Are there any noticeable differences in the trend of population growth in the following countries? Why?
Rwanda (bar)
Populations
Year
Algeria (bar)
Populations
Year

#2: Pixel Equations

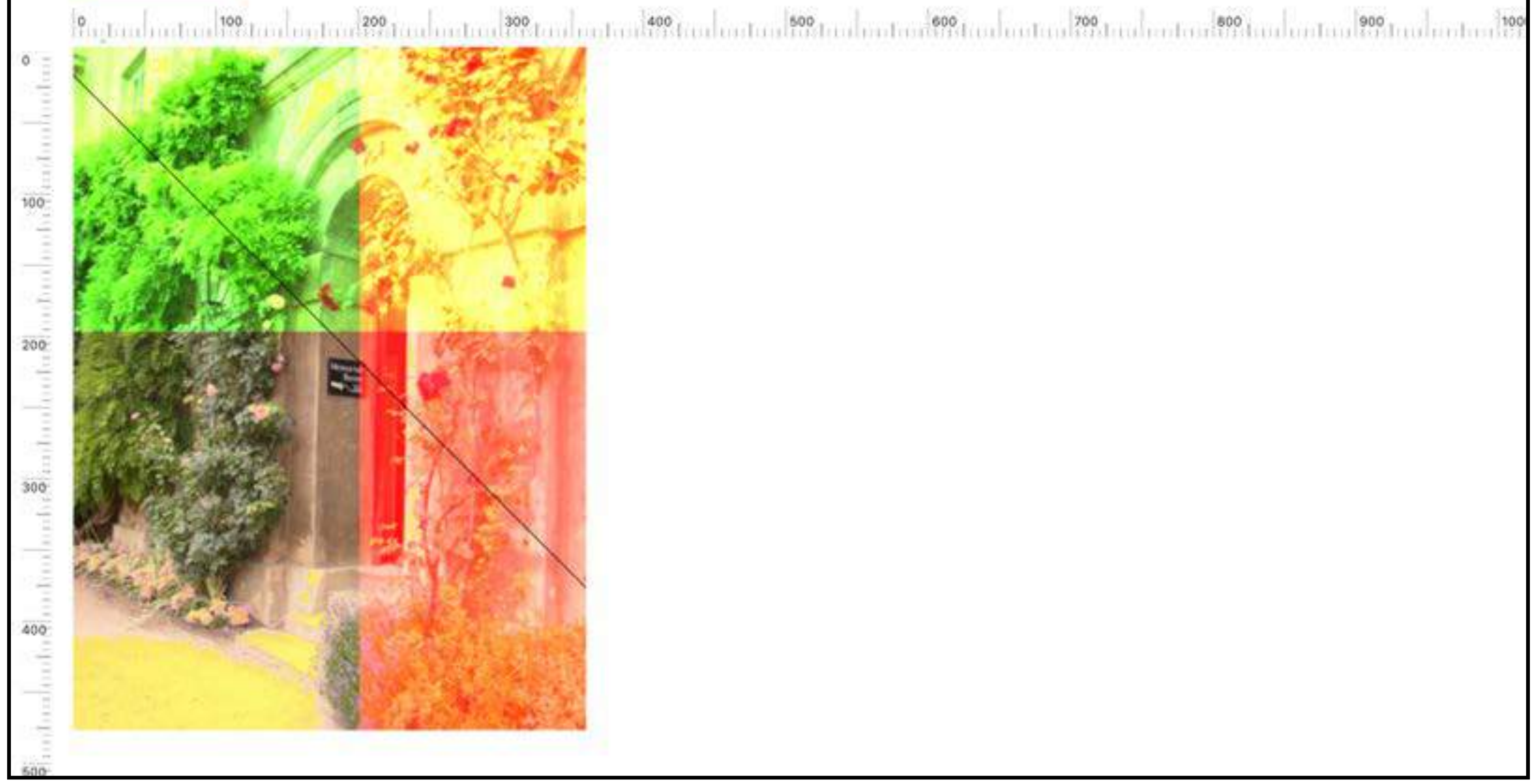
For Math and Engineering classes

If this is true Si esto es cierto	Set Red Asignar Rojo	Set Green Asignar Verde	Set Blue Asignar Azul
x > 200	255		
y < 200		2 * green	
blue > 200			blue / 2
x = y - 20	0	0	0

Step 3: Run Equations

Result Picture Appears Here:

Show Result



Pixel Equations

Select your preferred language

◉ English

Idioma/Language

Step 1: Pick your input picture

Which picture would you like to use?

◉ File named: arch.jpg 

○ File named: Bayamon.jpeg 

○ File named: beach.jpg 

○ File named: dog.png 

○ File named: san-juan.jpeg 

○ File named: TSM-Map.png 

○ File named: detroit.jpg 

○ File named: DetroitSkyline.jpg 

which will select all pixels where the `x` coordinate is greater than the `y` coordinate.

Then write equations for how to change red, green, and blue (rojo, verde, y azul) for the selected pixels. You can invert each color by subtracting from 255 (e.g., set red/rojo to `255-red` (o `255-rojo`)).

If this is true **Si esto es cierto**	**Set Red** **Asignar Rojo**	**Set Green** **Asignar Verde**	**Set Blue** **Asignar Azul**
x > 200	255		

Step 3: Run Equations

Result Picture Appears Here:

Show Result

set red/rojo to `255-red` (o `255-rojo`)).

If this is true Si esto es cierto	Set Red Asignar Rojo	Set Green Asignar Verde	Set Blue Asignar Azul
x > 200	255		
y < 200		2 * green	

Step 3: Run Equations

Result Picture Appears Here:

Show Result

Then write equations for how to change red, green, and blue (rojo, verde, y azul) for the selected pixels. You can invert each color by subtracting from 255 (e.g., set red/rojo to `255-red` (o `255-rojo`)).

If this is true Si esto es cierto	Set Red Asignar Rojo	Set Green Asignar Verde	Set Blue Asignar Azul
x > 200	255		
y < 200		2 * green	
blue > 200			blue / 2

Step 3: Run Equations

Result Picture Appears Here:

Show Result

#3: Counting Sheets

Elise Lockwood: Teaching combinatorics with Python

Student challenge:
How many arrangements do you expect to get from the letters in the word ROCKET?

- No repeated letters
- Order matters

Why do you think that? How will the list of outcomes be structured?

```
arrangements = 0
People = ['R','O','C','K','E','T']

for p1 in People:
    for p2 in People:
        if p2 != p1:
            for p3 in People:
                if p3 != p1 and p3 != p2:
                    for p4 in People:
                        if p4 != p3 and p4 != p2 and p4 != p1:
                            for p5 in People:
                                if p5!=p4 and p5!=p3 and p5!=p2 and p5!=p1:
                                    for p6 in People:
                                        if p6!=p5 and p6!=p4 and p6!=p3 and p6!=p2 and p6!=p1:
                                            arrangements = arrangements + 1
                                            print(p1, p2, p3, p4, p5, p6)
print(arrangements)
```

"Reinforcing key combinatorial ideas in a computational setting: A case of encoding outcomes in computer programming," 2021, Journal of Mathematical Behavior, Lockwood and De Chenne

Counting Sheet Interactive Tool

First time here?

Try a code snippet:

2: Shirts and pants

Counting Sheet:

col1	col2	col3	col4	col5	col6
tee,polo,swea ter	jeans,khaki				

Generate ☐ Chart Output

Results:

tee jeans
tee khaki
polo jeans
polo khaki
sweater jeans
sweater khaki

Is this computing education?

WHAT ARE STUDENTS LEARNING HERE?

Rich, Strickland, Binkowski, Moran, and Franklin (ICER 2017) asked the question:

What's the starting place for K-8 CS learners?

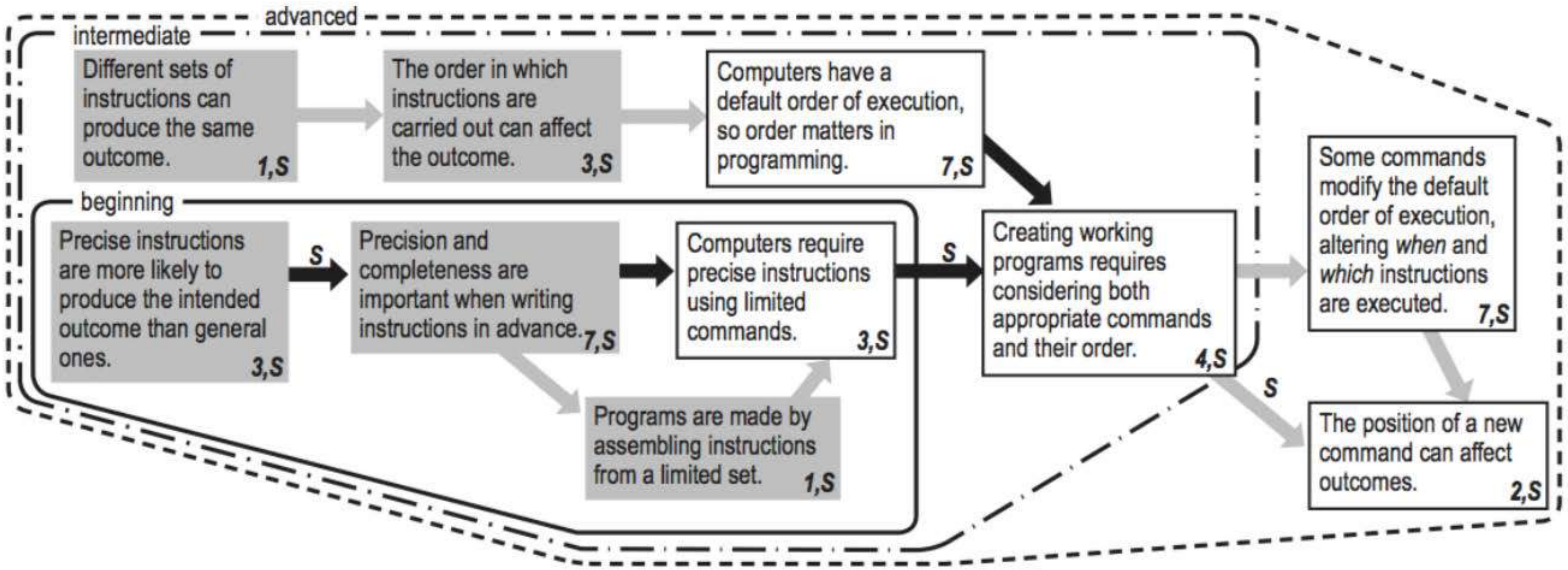


Figure 3: Sequence learning trajectory.

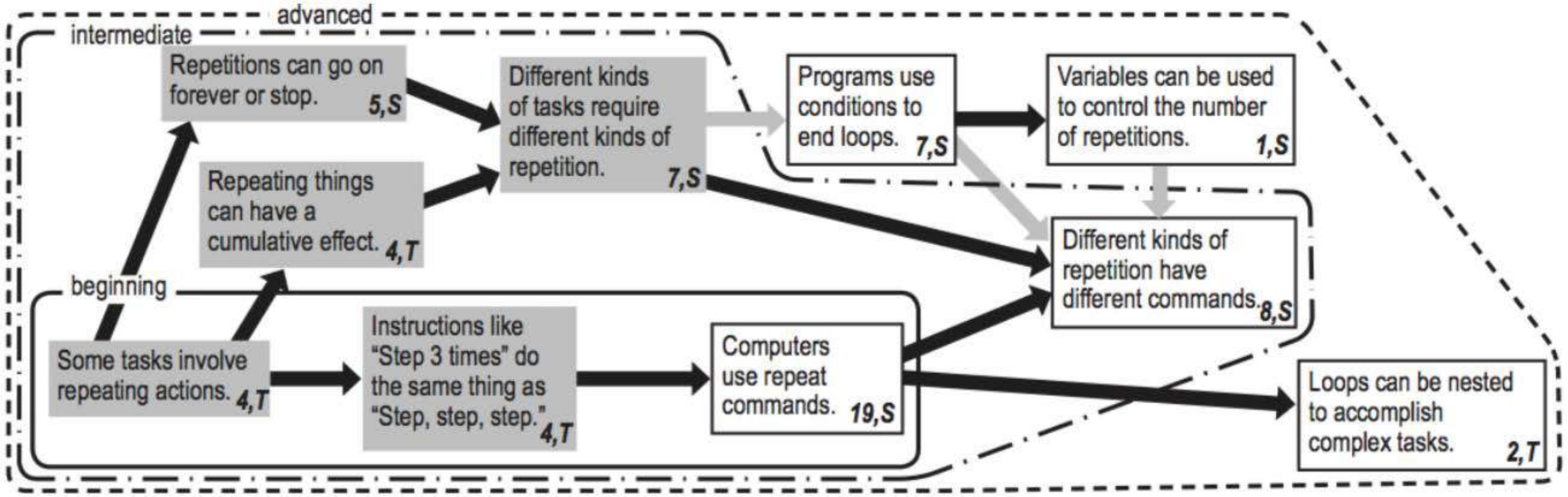


Figure 4: Repetition learning trajectory.

Proposed:
What comes first when learning programming?

1. Precision and completeness are important when writing instructions in advance.

2. Different sets of instructions can produce the same outcome.

3. Programs are made by assembling instructions from a limited set.

4. Some tasks involve repeating actions.

5. Programs use conditions to end loops.

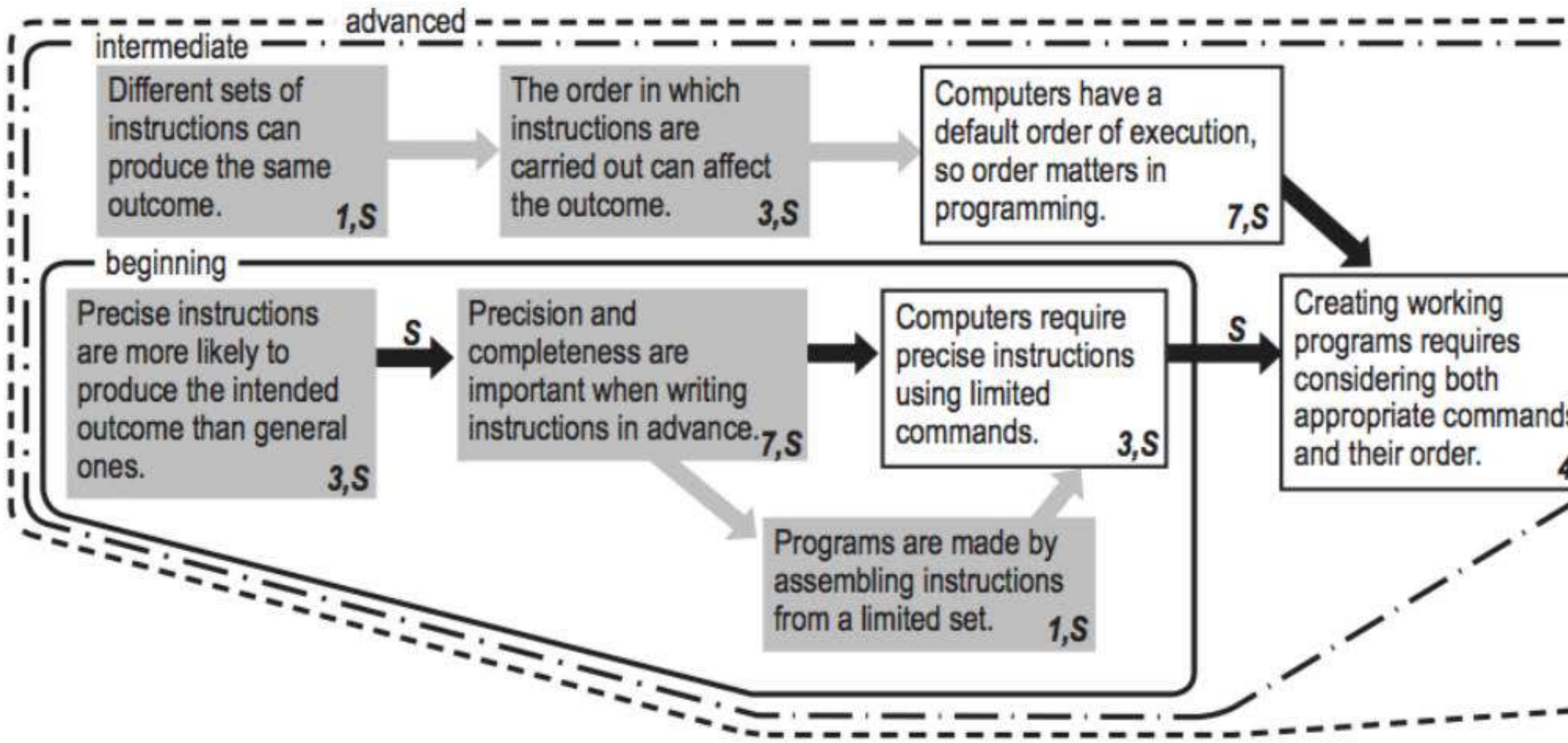


Figure 3: Sequence learning trajectory.

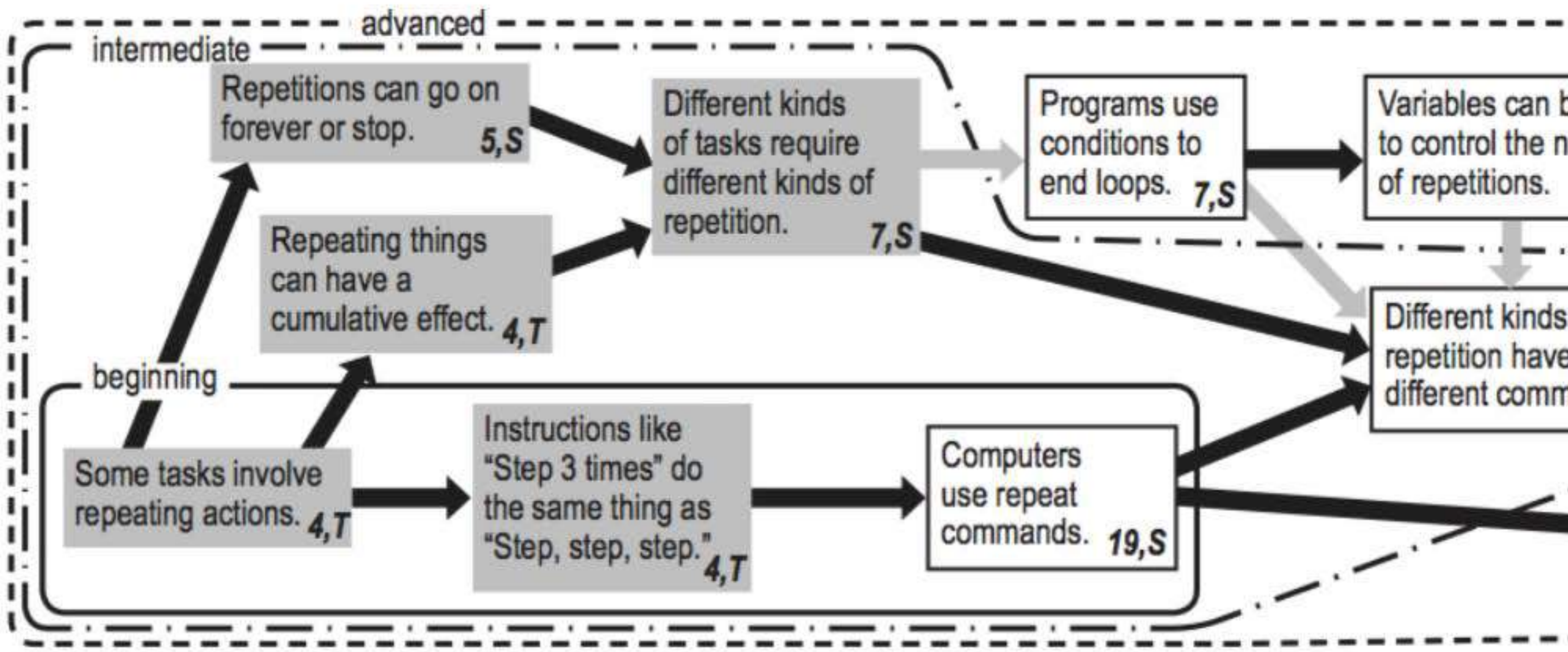
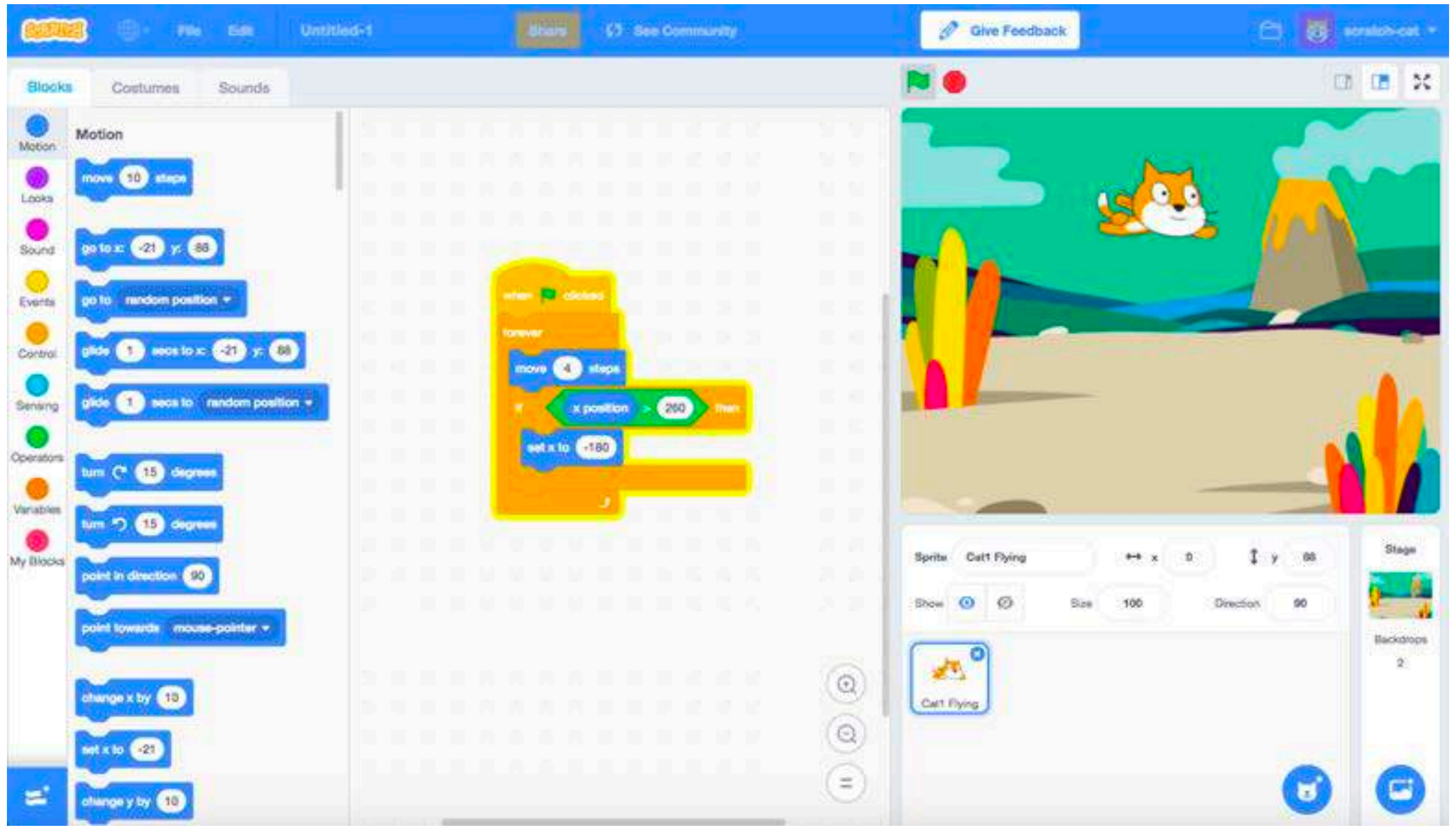


Figure 4: Repetition learning trajectory.

Scratch fluency doesn't need that whole list

- Over 60 million users.
- Most Scratch projects are stories that use…
  - Only Forever loops
  - No booleans
  - Just movement and sequence.

There is *expressive* power in even a subset of CS.



Bootstrap:Algebra doesn't use all of that list

- Improves learning in algebra
- Students do not code repetition.
- Functional

BOOTSTRAP
Equity • Scale • Rigor

There is *learning* power in even a subset of CS.

Unit	Game Feature	Programming Concept	Math Concept
1	locating elements on screen	expressions, Circles of Evaluation	coordinates
2	creating text and images	string and image operations	domain, range, kinds of data
3–5	making moving images	defining functions, examples	multiple function representations: as formulas and as tables
6	determine when game elements are off-screen	Booleans and Boolean operators	inequalities
7	responding to key-presses	conditional	piecewise function
8	collision detection	(nothing new)	Pythagorean Theorem
9	polishing games for presentation	code reviews	explaining math concepts to others

Figure 1: Curriculum structure: each unit introduces game, programming, and math concepts in parallel.

Schanzer, Fisler, Krishnamurthi, Felleisen, 2015

Learning challenges that our teachers face

And probably our students, too.

- ***Intermediate representations***:
  - Much of computing involves use of a notation (HTML, programs) that is interpreted by a computer for a final result (web page, program execution).
- **Debugging**:
  - The computer only interprets your notation — it does not know your intention.
    When the interpretation does not match what you intended for the result, you will have to debug.

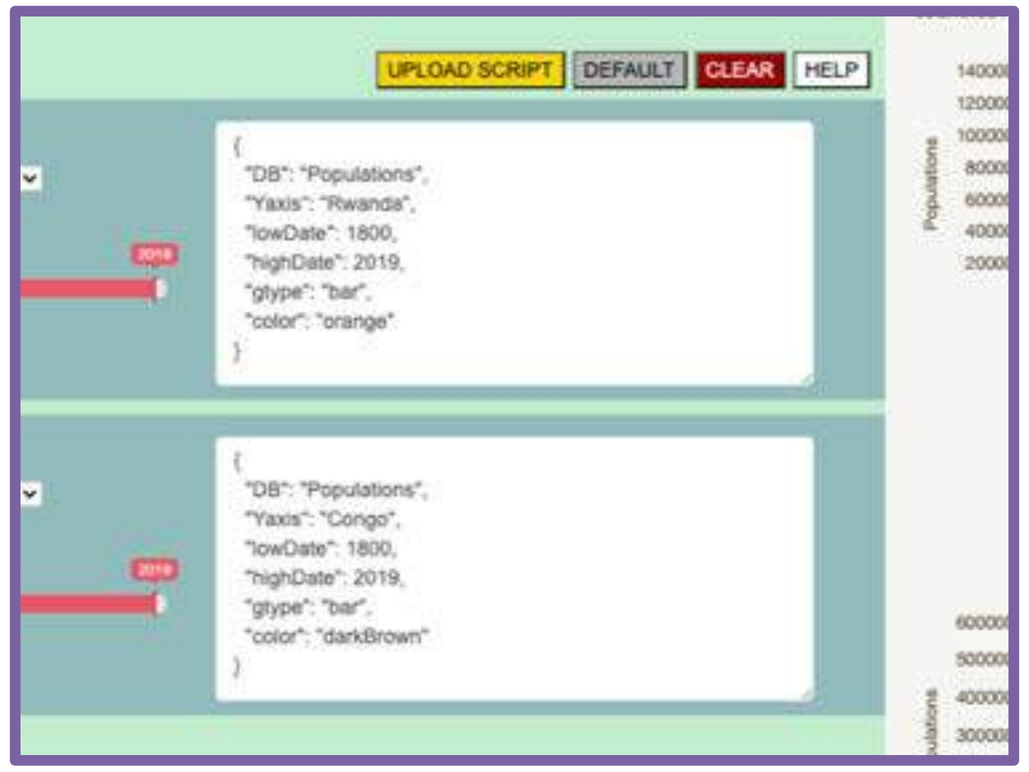
REPEATING THE BOTTOMLINE:

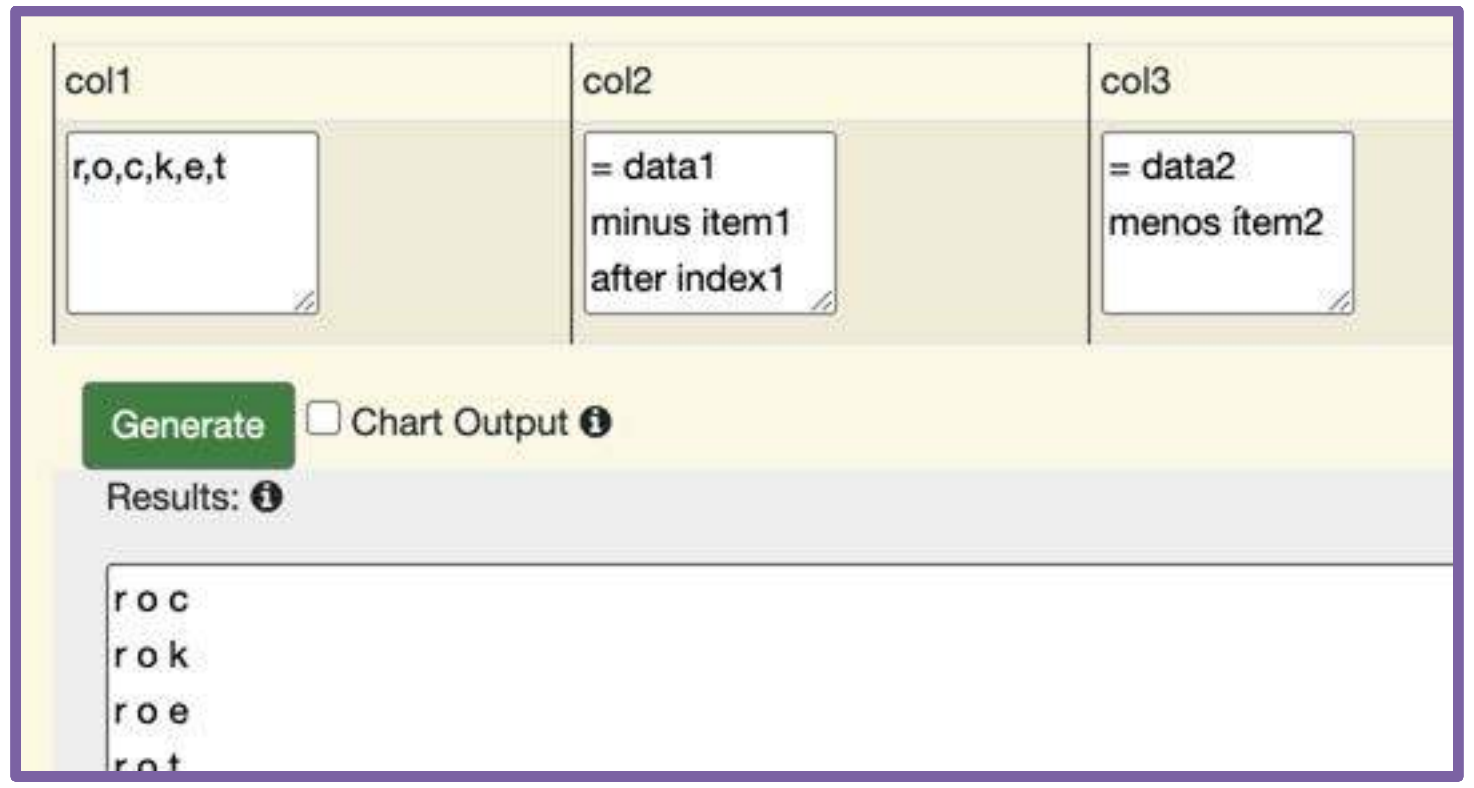
THE MAJORITY OF SECONDARY SCHOOL STUDENTS IN THE US AND ENGLAND HAVE NEVER SEEN COMPUTER SCIENCE

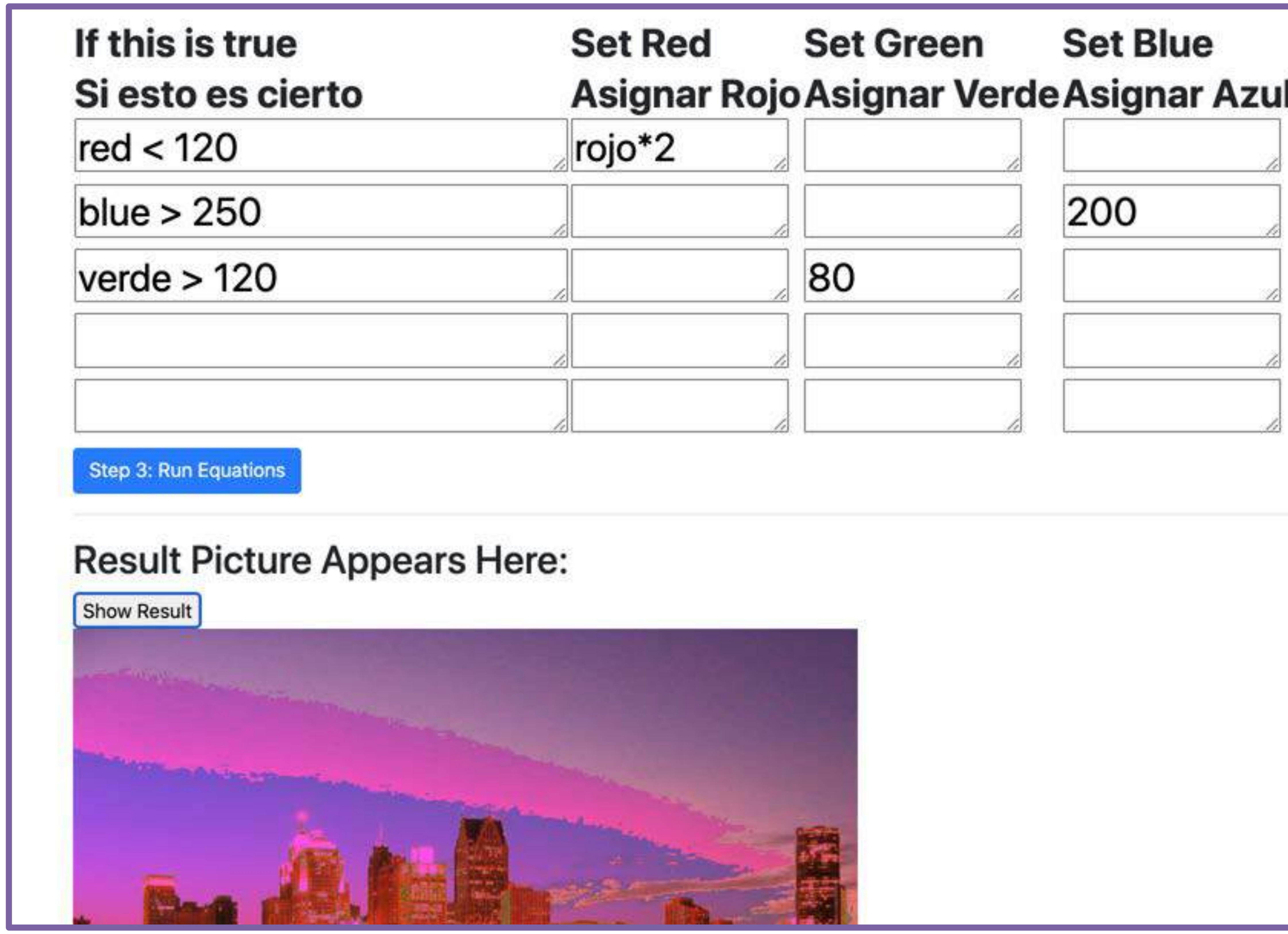
We don't know much about teaching *all* students about computing

This too is Programming

A place to learn about intermediate representations and debugging.
Useful tools in social studies, mathematics, and engineering.







We are developing Teaspoon languages with English and Spanish keywords.

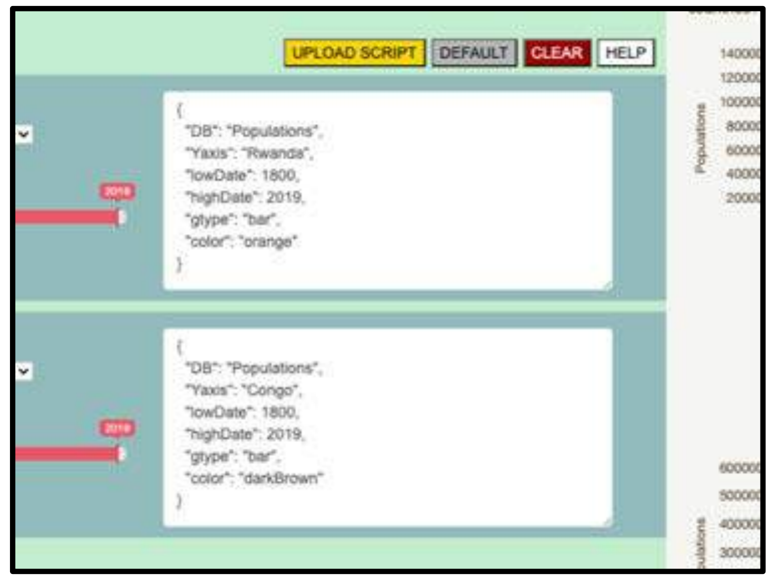
2/3 of the world does not speak English.

Reaching **everyone** requires new languages and tools

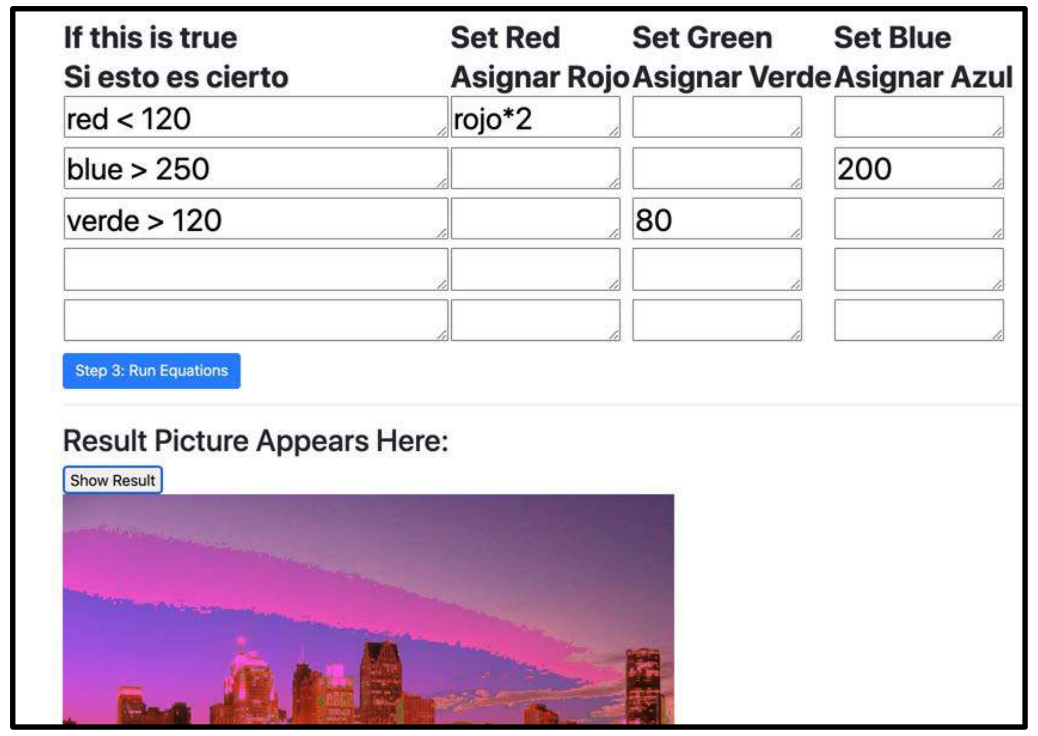
Teaspoon Languages as a CS for All Strategy

- **Hour of Code**: One hour of a Turing-complete programming language every year.

- **Teaspoon Languages:** One to three little languages in every social studies, mathematics, and language arts class.

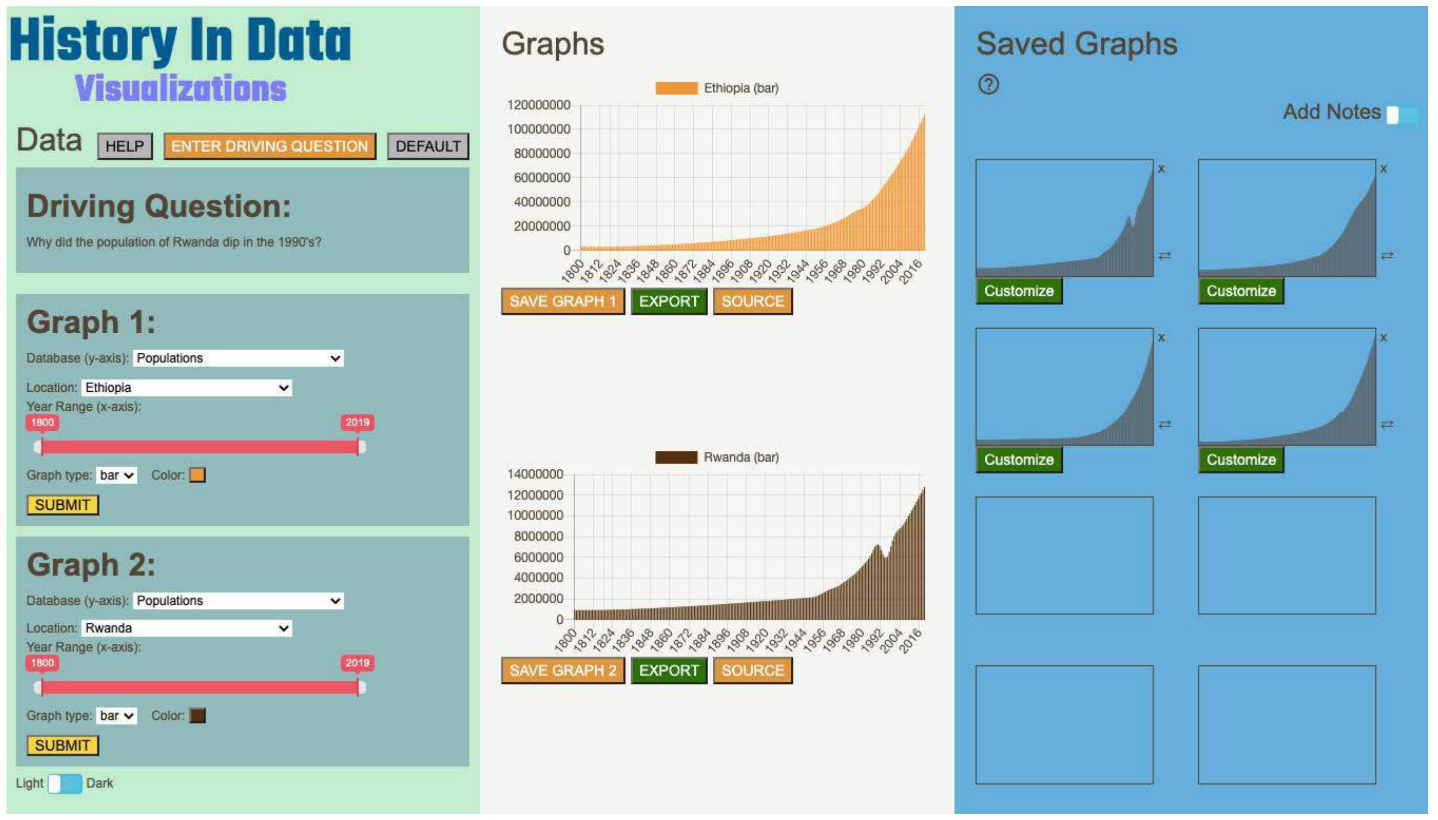


- Which results in more retained and transferrable CS learning?
  Which creates more of a school culture about using programming across disciplines?



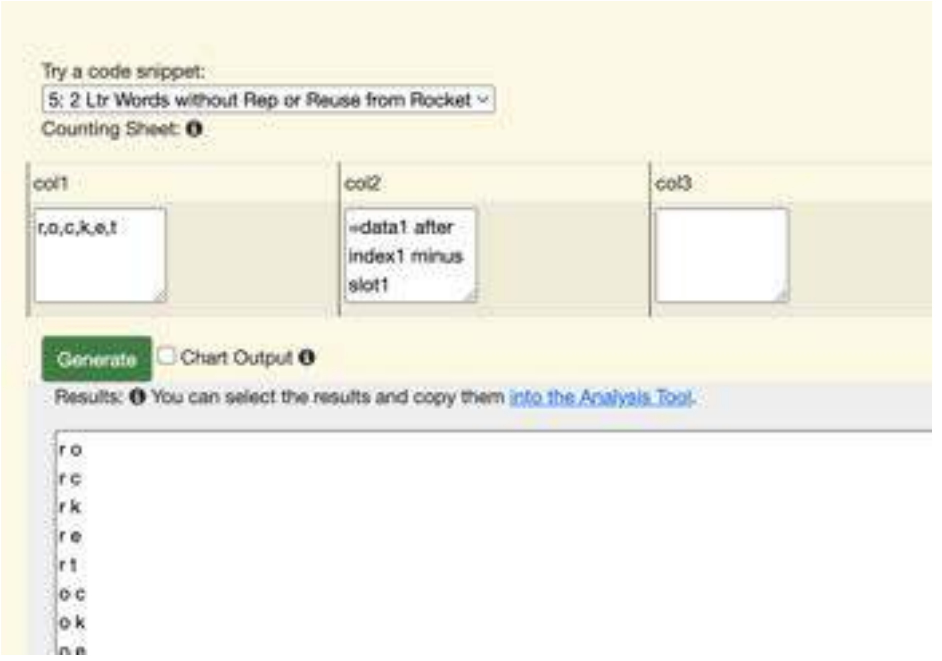
Time to Play with *PROTOTYPES*

- For history: DV4L



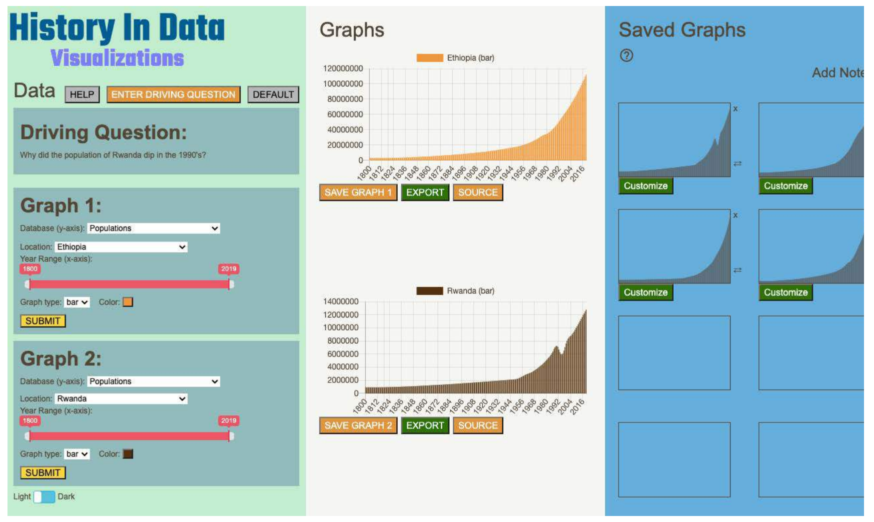
- For Mathematics/Engineering: Pixel Equations

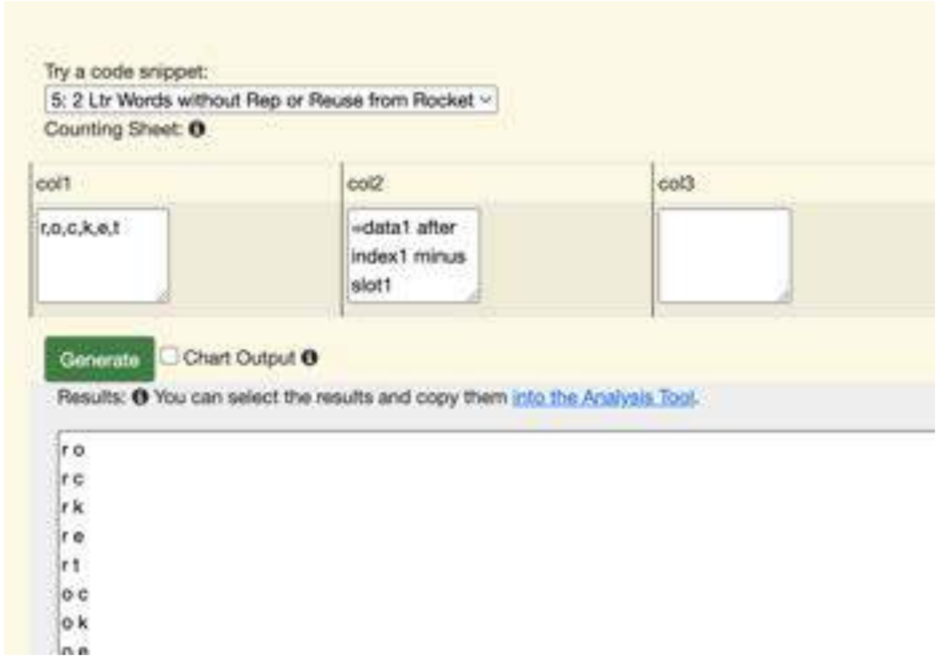
- For Mathematics/Counting: Counting Sheets



Questions to Think About

- What would it take to get other-than-CS teachers in your schools to try a Teaspoon language?
- Do you see students struggling with the fundamental issues of intermediate representations and the left side of the learning trajectories?
- How would you improve Teaspoon languages? For what tasks should we be developing new Teaspoon languages?





Collaborators on This Work

- Bahare Naimipour, Tamara Nelson-Fromm, Emma Dodoo, Tammy Shreiner, Elise Lockwood, Adaline de Chenne.
- Undergraduate researchers: Aryan Bannerjee, Alexandra Rostkowycz, Erin Shi, Brandon Geng, Jessica Zhang, Ben Steinig, Kashmira Reddy, Kristen Taurence, Angela Li, Derrick White, Jessie Houghton.

- http://computinged.wordpress.com
- http://guzdial.engin.umich.edu

Some of this material is based upon work supported by the National Science Foundation under Grant No. 2030919. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Thank you!

SPARE SLIDES