

# Teaching for a Data-Driven Future

## Intentionally Building Foundational Computing Skills

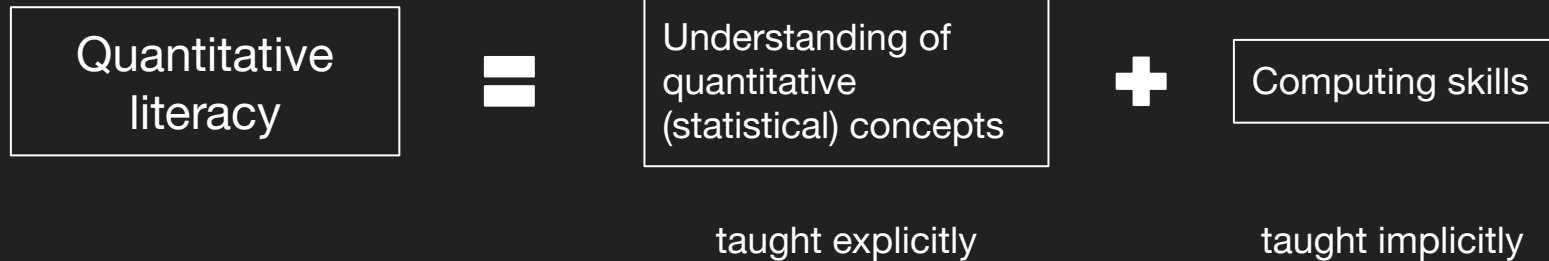
Amy L. Johnson & Rebecca D. Gleit  
Stanford University  
ASA 2020

# Roadmap

1. The challenge
2. Our solution
3. Foundational computing concepts
4. Our workshop
5. Key takeaways

# The challenge

Sociology curricula need to emphasize applied data skills.



# Our solution

Teach computing skills explicitly.

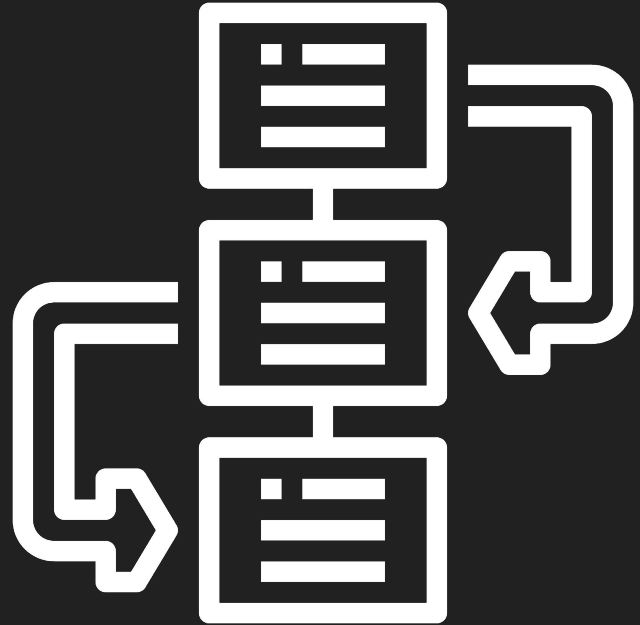
1. Better prepare students for post-graduate opportunities
2. Equip students to solve real social problems
3. Ensure equity by moving computing skills out of the hidden curriculum

Three foundational computing concepts

# 1. Workflow

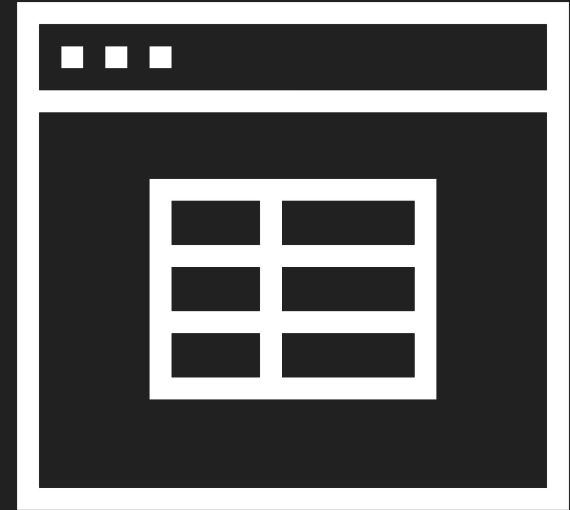
Storing data and files on a computer in a well-organized and accessible way.

Writing reproducible, efficient, and “clean” code. Focus on coding as communication.



## 2. Data storage

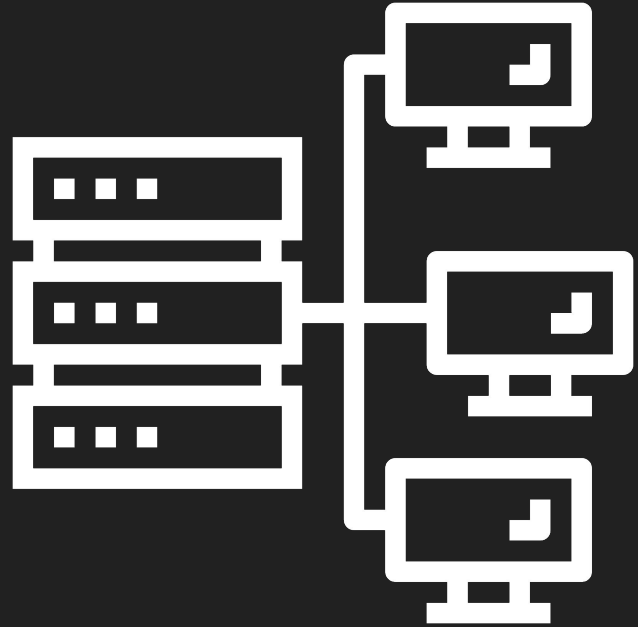
How statistical software programs understand and store data.



### 3. Programming logic

Subsetting: the process of dividing the full data set into different pieces using logical statements

Functions: tasks the software program knows how to do, and are used to manipulate (subsets of) data





Our workshop

# 2-hour introductory workshop

## First hour:

Interactive lecture: coding “by hand”

***Without*** computers

Focused on concepts

Materials:

<https://bit.ly/SoftwarePres>

<https://bit.ly/SoftwareHandout>

## Second hour:

Self-directed: students work through step-by-step guide

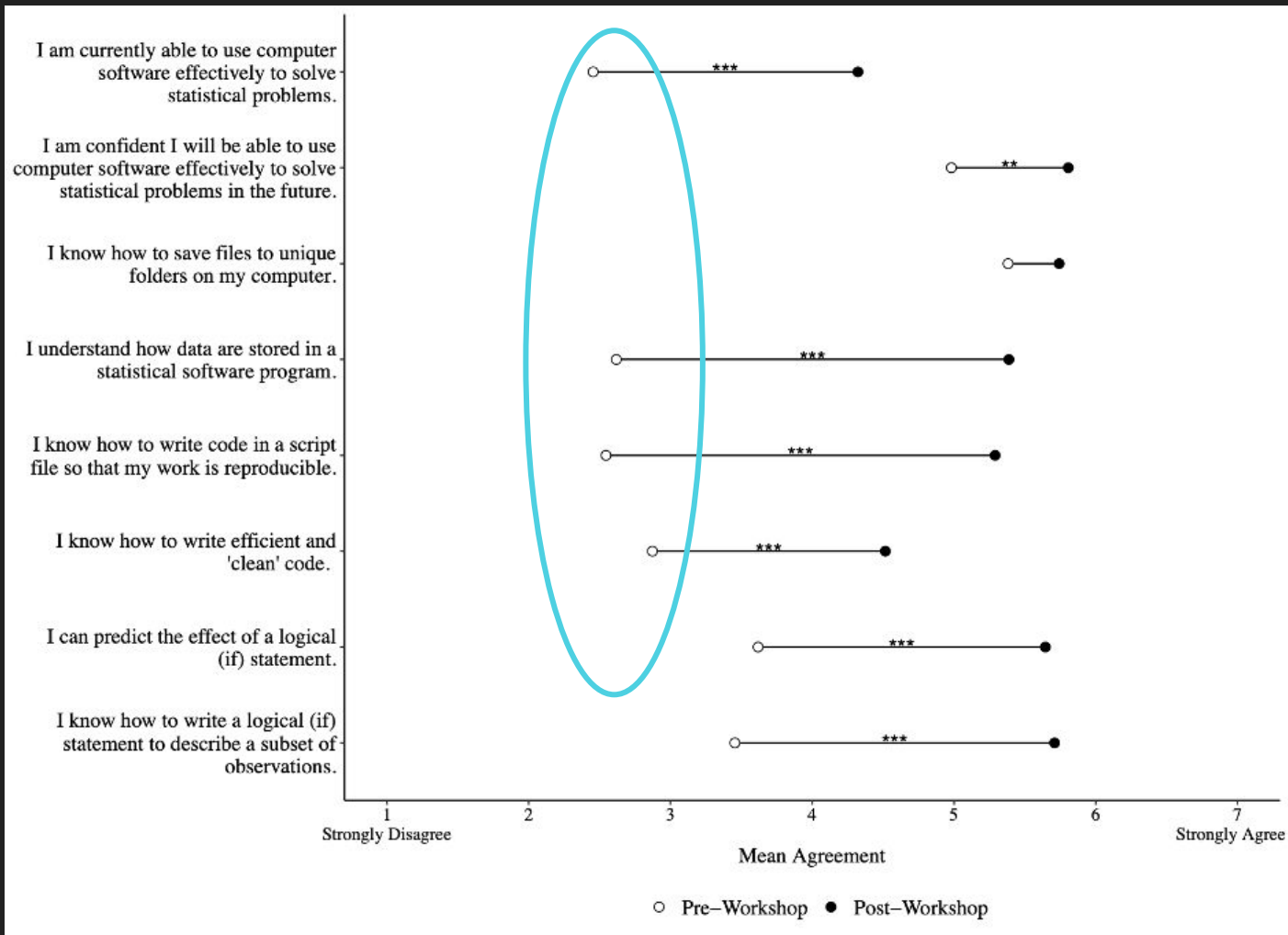
Using computers

Focused on implementation

Materials:

<https://bit.ly/SoftwareSelfDirected>

Fig 1. Pre- and post-workshop self-rated ability



# Key takeaways

- Merely using statistical software is not enough to ensure that students develop computing skills.
- The introduction to computing is crucial, but instructors should explicitly teach computing skills throughout their quantitative courses.
- Moving computing skills out of the “hidden curriculum” and into the formal course agenda creates a more equitable learning environment, more explicitly equips students with transferable data skills, and prepares them for participation in an increasingly computational world.

# Tangible suggestions



→ Organize folders by project, and use sub-folders

→ Be clear about what a “working directory” is

→ Provide students with a code template

→ Lead by example



→ Explain the matrix-like structure of data frames: rows as units, columns as variables

→ Explain what variables *are*, including different types of variables and how they’re stored

→ Use data that students created



→ Use visuals, e.g. Venn diagrams, function diagrams

→ Practice coding “by hand”

→ Translate between code and English statements