Exploring How Undergraduate Students Engage in Computational Thinking with Data

Alyssa Hu
Pennsylvania State University

Neil J. Hatfield
Pennsylvania State University

Matthew D. Beckman
Pennsylvania State University

Modern statistics education requires that we support students in building powerful and productive ways of computational thinking. In this paper, we seek to understand the ways of computational thinking that undergraduate students employ as they engage with data. To address this question, we administered task-based interviews with three participants using the R programming language. Problem-solving approaches focusing on formatting data and efficient coding emerged as early aspects of student thinking. We are still reviewing interview transcripts and intend to compare our findings with existing frameworks from the literature, working towards a framework highlighting beneficial ways of thinking. This work is part of a larger study with additional tasks and additional individuals with varying levels of experience and expertise.

Keywords: computational thinking, data, statistics education, student thinking

The Guidelines for Assessment and Instruction in Statistics Education (GAISE) College Report endorsed several recommendations for introductory statistics courses, such as “use technology to explore concepts and analyze data”, “integrate data with a context and purpose”, and “teach statistics as an investigative process of problem-solving and decision-making” (Carver et al., 2016). These require that instructors support students in building powerful and productive ways of computational thinking.

Wing (2006)’s seminal paper on computational thinking emphasized that it is a “fundamental skill for everyone” that “involves solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science” (p. 33). Aho (2012) defined computational thinking as “the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms” (p. 832). Lockwood et al. (2016) built upon Aho’s work, by studying the role of such thinking in mathematics through interviews with mathematicians. They arrived at the working definition “a logical, organized way of thinking used to break down complicated goals into a series of (ordered) steps using available tools”, which they termed “algorithmic thinking” (Lockwood et al., 2016, p. 1591). Shute et al. (2017) considered research in K-12 and higher education settings to define computational thinking as “the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts” (p. 151). These definitions may appear overly broad, so we discuss existing frameworks to ground them.

Background

Computational thinking cuts across multiple disciplines including computer science, statistics, and mathematics; further, computational thinking appears at all levels of education. Working from a computer science education perspective, Barr and Stephenson (2011) discussed the core computational thinking elements that emerged during a Thought Leaders meeting for K-12 implementation organized by the Computer Science Teachers Association and the International Society for Technology in Education. Focusing on a particular environment, Brennan and Resnick (2012) studied young programmers (ages 8-17) through workshops and online project portfolios. To study the effect of age and gender on the development of computational thinking skills, Atmatzidou and Demetriadis (2016) conducted robotics training seminars in Greek junior high (age 15) and high school (age 18) classrooms. Weintrop et al. (2016) focused on creating a taxonomy of computational thinking practices to incorporate in math and science curriculums at the high school level. Lastly, Shute et al. (2017) proposed their own model after a thorough literature review and extensive discussion of the four previously mentioned works.

Despite these different contexts and methods, common themes emerged among each set of researchers' frameworks, speaking to the widespread relevance and applicability of computational thinking. One such common theme is the role of data. Barr and Stephenson (2011) discuss data collection, data analysis, and data representation as part of the core elements and capabilities of computational thinking. Brennan and Resnick (2012) refer to data as a key concept with regard to the storage, retrieval, and updating of values within computing environments. Within their taxonomy, Weintrop et al. (2016) consider five data practices– collecting data, creating data, manipulating data, analyzing data, and visualizing data. Shute et al. (2017) referenced data collection and analysis. Thus, we see that anticipating and planning data wrangling as early and vital components to computational thinking. Additional commonalities appear in Table 1.

Table 1. Additional commonalities with accompanying references.

Themes	References
Decomposition	Barr and Stephenson (2011); Atmatzidou and Demetriadis (2016); Shute et al. (2017)
Algorithms	Barr and Stephenson (2011); Atmatzidou and Demetriadis (2016); Shute et al. (2017)
Parallelization	Barr and Stephenson (2011); Brennan and Resnick (2012)
Simulation	Barr and Stephenson (2011); Weintrop et al. (2016)
Generalization	Atmatzidou and Demetriadis (2016); Shute et al. (2017)
Modularity	Brennan and Resnick (2012); Atmatzidou and Demetriadis (2016); Weintrop et al. (2016)
Debugging	Brennan and Resnick (2012); Shute et al. (2017)
Iteration	Brennan and Resnick (2012); Shute et al. (2017)
Abstraction	All frameworks listed

Theoretical Perspective

When discussing these five computational thinking frameworks, we find utility in thinking about them through the lens of Harel's (2007) DNR determinants. Harel (2007) proposed the DNR system for curriculum development and instruction which consist of premises, determinants, and instructional principles (specifically: Duality, Necessity, and Repeated reasoning). Harel describes the determinants of *mental act* (actions performed in constructing knowledge), *way of understanding* (product or outcome of a mental act), *and way of thinking* (character or feature of a mental act). An example provided in the paper illustrates how proving is a mental act, a proof is a way of understanding, and a proof scheme is a way of thinking. Within the present study, the code that a participant writes to import data would reflect their way of understanding data importing while their imagery for and anticipations of loading data into an environment would reflect their way of thinking about data importing. With this backdrop in mind, we define computational thinking as *the ways of thinking that individuals employ when using computational tools to problem-solve efficiently and effectively.*

Many of the common themes (see Table 1) and our own definition involve the mental act of problem-solving. Barr and Stephenson (2011) also discuss some dispositions and pre-dispositions (e.g., “confidence in dealing with complexity”) and classroom culture (e.g., “increased use of computational vocabulary”) that would facilitate productive ways of computational thinking. These appear to be outcomes of problem-solving. Similarly, Brennan and Resnick’s (2012) computational thinking concepts (e.g., “loops”) and computational thinking perspectives (e.g., “questioning”) appear to be products of problem-solving. We consider these to be ways of understanding. Our research focuses on the question: **What are ways in which computational thinking appears as part of undergraduate students’ thinking as they work with data?**

Methodology

This project is part of a larger study regarding computational thinking for individuals of varying levels of experience and expertise. In this report, we narrow our focus to two tasks designed to probe attributes of computational thinking while participants engage and wrangle with complex data to achieve a defined purpose. While many computational tools exist, we focused on the R programming language and its related software (e.g., R Studio).

Participants

We interviewed three undergraduate students enrolled in a data science course focusing on statistical reasoning and computation, which has a pre-requisite of an introductory R course. The task-based interviews (Goldin, 2000) were recorded over Zoom. Students also submitted their R file in which they worked. We refer to the students as S1, S2, and S3.

Tasks

The two tasks are based on the 2021 American Time Use Survey dataset (ATUS; U.S. Bureau of Labor Statistics, 2022). Participants are provided with: (a) the raw Activity Summary data file from the 2021 ATUS, (b) a simplified data dictionary we created for demographic variables in the data file, (c) an activity lexicon to describe the activity variables in the data file, and (d) task prompts. The ATUS data was selected due to the topic requiring minimal background knowledge and its relevance as real (not simulated) data with practical applications. Both tasks follow a similar format, where we present a data visualization, ask participants regarding their observations of the chart, and then observe the participants as they explore and analyze the data to calculate a particular estimate that is displayed using the ATUS data.

Task 1 prompt. Consider the visualization below (Figure 1) made by the U.S. Bureau of Labor Statistics which tells us the average hours per day spent in selected activities. “Average per day, total” is selected. What does this chart tell you about the average hours per day, total, spent on personal care, including sleep? Would you walk me through how you would recreate this estimate using the atussum_2021.dat file?

Task 2 prompt. We maintain the same format of Task 1, but the context of the problem is modified. In Task 2, both “Average per day, men” and “Average per day, women” are selected and compared in a data visualization. We are interested in how participants address the coding for sex in the dataset and how they adapt or modify their previous solution to Task 1. (We omit the full prompt and figure in this report for brevity.)

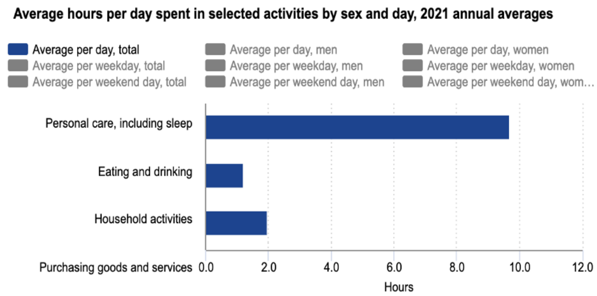

Figure 1: Task 1 Data Visualization from U.S. Bureau of Labor Statistics (truncated for space constraints)

Results

Preliminary analyses are currently underway. Below, we share some excerpts from participants and a couple of potential ways of thinking (pertaining to problem-solving approaches) that could be emerging.

Problem-Solving Approach: Ensure Proper Format of Data

All participants considered how to load data into RStudio. They used different coding environments; S1 used an R script, S2 used an R Markdown file, and S3 opted for an R Notebook file. S1 proceeded to directly write a *read.csv*() command and ran the line of code which provided a preview of the data frame in the console. S2 and S3 both used the data import wizard (GUI) of the RStudio IDE. Despite the two different ways of data importation, all students performed a visual inspection of the data frame; S1 checked post-import while S2 and S3 checked during import. They all commented on what they perceived to be correct format:

S1: So everything is already separated in columns.

S2: Let's see, and I see that there is a heading so "Yes", import that.

S3: The variable column headings are stuck as V1, V2, it goes on. So what I did was switch Heading to "Yes" and this makes it the proper headings.

In contrast to S2 and S3, S1 added in a couple of lines of code using the *complete.cases*() and *unique*() commands. It appeared that S1 had additional data cleaning procedures to apply regardless of the data context, based on the following conversation:

S1: That's usually what I do first, I think.

Interviewer: And can you remind me again, what these 2 lines of code will do?

S1: So, this, the first one, is to make sure all, everything is completed. There's nothing empty, nothing like the rows. There's no one with missing information. And then, the second one is to make sure there's nothing repeated.

Interviewer: Gotcha. And you mentioned that you, you said that this is sort of what you normally do. So are these things that you check every time you code, normally?

S1: I think so, to make sure everything is more clean.

Problem-Solving Approach: Find Efficient Way to Write Code

When the interviewer asked participants to talk about their approach for Task 2, all participants attempted to find the average hours for men and average hours for women using two distinct code blocks. They all expressed a desire for alternative ways to write the code:

S1: Maybe there is way to do female on the same dataset as the male and then write a code to do the rest. I'm just not sure how I'd write that, but I think, the easiest way for me to do [it] would be [to] do separately, but I'm sure there's probably a way to do it together.

S2: If I want to do it together, how would that work? So then I'm thinking, like, okay, I, the very beginning part's the same. Uh, I'm just thinking about when I plot it, how am I gonna be able to distinguish between male and female and how I want it, I guess. Like, your code running is very important but also I think having it be very concise and clean is also pretty important.

S3: So, there probably is a way to do it within one data set, but it's just the first thing that came to mind.

Based on their terminology ("do it together" and "do it within one"), all three participants had considerations of reducing redundancy and increasing efficiency. S2 specifically verbalized the reason for doing so, including anticipation of future analysis (plotting) and what they considered to be best practices of coding.

Discussion

We have outlined some initial ways of thinking that undergraduate students possess and will continue to analyze our data. Our next steps will be to consider the points of commonality with (and departure from) existing frameworks. This would involve additional coding of the data based on these frameworks, and it may be useful to note the frequency and length of time that participants spend in each unique code.

We note some planned future work as this work is part of a larger study. One direction is collecting data from individuals with additional experience and expertise, including graduate students, faculty, and industry professionals. This would allow us to consider a new research question: What similarities and differences do we observe among individuals who self-identify along the novice-expert continuum? We will also consider how an individual may move along the expert-novice continuum in terms of their data-ing and computational thinking skills. Another direction is analyzing participant responses to several additional tasks, including: (a) creating a visualization, (b) brainstorming their own question about the dataset, (c) discuss coding challenges and review code, and (d) verbalize how they would approach a new prompt about the dataset. The purpose of (a) and (b) are to highlight how and what we can communicate with data. We hope to spark a discussion regarding effective problem-solving with (c) and how computational thinking transfers across different contexts with (d).

A key outcome from our study will be understanding the role of computational thinking, in conjunction with interrogating data sources, in reshaping the statistics curriculum. In creating our framework, we will highlight concepts and skills that we might want to foster and support as educators. This would support the generation of hypothetical learning trajectories, which can then inform classroom interventions and expectations while engaged in data exploration, analysis, and communication. In addition, we can reflect and learn from participant feedback regarding the tasks we created, which can inform the way we design future tasks and instruction in statistics, data science, and related courses. A possible extension in the future would be to develop an assessment to measure student development of computational thinking skills.

Intended Questions for Audience

1. What similarities or differences in students' thinking have you noticed in your classroom experience compared to our (preliminary) results?
2. What are your thoughts on or suggestions regarding our planned future work?
3. What suggestions do you have for supporting students in further developing their computational thinking?

References

Aho, A. V. (2012). Computation and computational thinking. *The computer journal*, *55*(7), 832-835.

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75*, 661-670.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?. *ACM Inroads*, *2*(1), 48-54.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, pp. 25).

Carver, R., Everson, M., Gabrosek, J., Horton, N., Lock, R., Mocko, M., Rossman, A., Roswell, G. H., Velleman, P., Witmer, J., & Wood, B. (2016). Guidelines for assessment and instruction in statistics education (GAISE) college report 2016.

Goldin, G. A. (2000). A scientific perspective on structured, task-based interviews in mathematics education research. In *Handbook of research design in mathematics and science education* (pp. 517-545).

Harel, G. (2007). The DNR system as a conceptual framework for curriculum development and instruction. In R. A. Lesh, J. J. Kaput, & E. Hamilton (Eds.), *Foundations for the Future in Mathematics Education* (pp. 21). Erlbaum.

Lockwood, E., DeJarnette, A. F., Asay, A., & Thomas, M. (2016). Algorithmic Thinking: An Initial Characterization of Computational Thinking in Mathematics. *North American Chapter of the International Group for the Psychology of Mathematics Education.*

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research review*, *22*, 142-158.

U.S. Bureau of Labor Statistics. (2022). *American Time Use Survey—2021 Microdata Files* [Data file and code book]. https://www.bls.gov/tus/data/datafiles-2021.htm

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology*, *25*, 127-147.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.