Using Computational Thinking for Data Practices in High School Science

Erin Peters Burton, Peter Rich, Timothy Cleary, Stephen Burton, Anastasia Kitsantas, Garrett Egan, and Jordan Ellsworth

This material is based upon work supported by the National Science Foundation under Grant No. 1842090. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the National Science Foundation.

Students often need to obtain, organize, clean, and analyze data in order to draw conclusions about a particular phenomenon (e.g., why tidal heights change). When conducting a science investigation in biology, chemistry, physics, or earth science, data can be collected by the student or can be provided to them via secondary data sets. This article demonstrates how computational thinking and data practices can be merged to develop more effective science investigation lesson plans.

What are Data Practices?

One of the science and engineering practices from the Next Generation Science Standards (NGSS), Using Mathematics and Computational Thinking, provides opportunities to help guide data practices in high school science investigations. Data practices are a suite of undertakings that students perform in order to make scientific claims from the evidence found in data. Weintrop and colleagues (2016) have identified five data practices that scientists engage in while doing investigations:

- Creating data: Generating data from tools or observation
- Collecting data: Gathering and recording data

- Manipulating data: Sorting, filtering, cleaning, normalizing, and combining data sets
- Visualizing data: Communicating results with a representation such as a graph or chart
- Analyzing data: Extracting meaning from a data set for the purpose of drawing conclusions

As teachers, we may recognize these practices as crucial processes that students must undertake in order to gather valid and reliable data and make strong conclusions about scientific phenomena. A challenge for many students, and particularly those with less experience with scientific practices, is making effective decisions regarding how to use data practices in an integrated way during science investigations.

Simply developing student knowledge about these data practices will not typically be sufficient for enabling students to use such practices. Students need guided practice and feedback, but can also benefit from learning how computational thinking practices can be used to explain the detail in the processes of data practices.

What is Computational Thinking?

Computational thinking (CT) is an approach to solving problems and designing systems that requires students to think recursively, reformulate problems to see them in a different light, model relevant aspects of problems, and use abstraction and decomposition in tackling complex problems. Wing (2006) defined CT to indicate a "thought process involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent." CT can be a useful addition to instruction as it

is a suite of complex processes that students can use to become skillful in data analysis in scientific investigations (Weintrop et al., 2016).

Because science and mathematics increasingly emphasize computation, introducing computational tools in the classroom can give a more authentic view of these disciplines (Augustine et al., 2005). Computational thinking practices can help students understand what to perform during data analysis in a way that reflects the scientific discipline.

There are several CT practices that are naturally linked with data analysis, such as decomposition, pattern recognition, abstraction, algorithm building, and automation. *Decomposition* is defined as the breaking down of a complex problem into less complex sub-problems. The specific approach to problem decomposition can vary, but the purpose is the same—to reduce the main problem into manageable steps or sub-problems.

For example, in order to better understand what causes tides, we might identify some variables that may affect tide height, such as the position of the moon or shape of the coast. In order to more systematically gather tide data, we might consider ways to measure the height of a tide, the format in which we would want that data, how often we want to measure tidal data, and where to place our instruments in order to get reliable information.

Another computational thinking practice is *pattern recognition*. Pattern recognition is defined as the identifying, clustering, and modularizing of steps, parts, or correlations that repeat or can be repeated. The primary purpose of identifying patterns is to cluster related parts of the problem by their recurring feature(s).

For example, in order to understand how a specific factor affects tide height, we might look for correlation (or pattern of behavior/relationship) between the cycle of the moon and tide

height. While manipulating collected data, we might notice an additional relationship (i.e., the nature of an observed pattern) between the shape of the water body and tide height, such as an increased correlation when controlling for time.

Abstraction is conceptualized as a process of identifying and organizing relevant information and removing unnecessary information. The purpose of abstraction is to clarify the problem and generate generalizable solutions, which is an essential skill for constructing models in science and engineering (Gilbert, 2004).

For example, in order to better understand the relationship between two variables such as tide height and position of the moon, we might remove outliers so that we can focus on the data points that are most coherent. When testing correlation with many different factors in our tides example, such as latitude or temperature, we might notice that some factors have minimal correlation with resulting tide height. We might then choose to omit that factor from consideration, given that it does not appear to have a meaningful impact on our observations or predictions.

Algorithm building is defined as the creation of a series of precisely defined steps or rules that leads to successful solutions to a problem. An algorithm, in simple terms, is an unambiguously defined process to address an initial question. It may involve the steps to collect certain data, the steps to analyze that data, or any other defined process. The steps of the algorithm should, if built correctly, lead to a correct solution of the problem every time, or within a known error chance. For example, in order to determine the factors that affect tide height, we might create a process of data collection and analysis to determine the extent to which location of the moon and shape of the water body impacts overall tide height. In order to predict future tide

heights, we might create a formula for calculating tide heights given specific patterns or measures identified.

Automation involves performing a procedure with little or no direct human interaction. This term typically refers to the use of machinery or computers to perform the automation. At this level of computational thinking, the goal is to outsource work so that it reduces or removes the requirement for direct human action in order to achieve the desired outcome. For example, with a defined formula for determining future tide height, we may automate the process of calculating the precise predicted tide height for the next 24 hours by programming the formula into a computer to calculate. Rather than collecting data by hand, we will use a machine or computer to measure a specific variable at predefined time intervals. Automation can occur even at small scales, allowing scientists to focus on data analysis rather than the minutia of repeated process.

For example, in order to reduce tedium in data manipulation and analysis (e.g., converting units of measure from imperial to metric), we may use a computer to store and manipulate our data in a quicker manner.

How Does Computational Thinking Help Student Data Practices?

Blending CT practices with data practices can provide students with a metacognitive guide for making decisions while engaged in scientific investigations. Based on an analysis of CT and data practices, we hypothesize that different combinations of CT practices are well-aligned or fit naturally with different data practices. Table 1 (see “On the web”) displays the crosswalk we developed to help teachers focus CT to the most relevant data practices. We felt

that not all CT practices and data practices align, and those areas are represented on the crosswalk by a blank space.

Table 1. *Crosswalk of computational thinking practices and data practices*

Data Practice	**Computational Thinking Practices**				
	Decomposition	Pattern Recognition	Abstraction	Algorithm	Automation
	What are the parts?	**Do any of the parts repeat or correlate?**	**What can we generalize?**	**What rules describe the process?**	**How can we automate the process?**
Creating Data	Identifying all components and processes necessary to collect the appropriate data to answer a research question is a critical first step in engineering any investigation.			Designing the investigation requires organizing the components and processes identified during decomposition into a logical order so that data will be repeatable.	Examining the investigation design algorithm to determine where automation may make the process more efficient.
Collecting Data		During data collection, pattern seeking is critical in determining if the design is generating appropriate data to answer the research question.	While collecting data, abstraction provides the process to identify confounding variables not previously identified as well as reduce any extraneous observations	As the process of data collection progresses, evaluating the algorithms being used is important to verify that the process is producing appropriate and unbiased data. New algorithms may be formulated to address any shortcomings of the previous design.	Automation during data collection must be monitored (at least initially) to verify that the automation tool is working within appropriate range of measured parameters or taking measurements in the appropriate time increments
Manipulating Data	An important part of generating descriptive statistics like mean, median, quartiles,	As the data are being organized, pattern recognition is	Abstraction can be used to help identify the extraneous material (noise)	Organizing data is algorithm dependent. The order of the data, the arrangement in	The widespread availability of computer programs allows for the

	etc. requires determining what data and descriptive statistics are necessary for answer the research question with the data collected.	helpful in determining if there are concerns from the data collection that might have influenced the resulting data beyond the parameters being tested	or attribute a reason for bad data	the tables, etc. are critical before descriptive statistics or visual representations can be created. These algorithms may be discipline specific as well.	automation of the data manipulation. However, it is important to recognize the potential pitfalls that the user might inadvertently cause (e.g., incorrect formulas, data incorrectly formatted, etc.)
Visualizing Data	Organizing your data for visualizing requires determining which components should and should not be displayed to answer the research question.	The primary role of visualizing data is to try to make patterns more explicit.	Abstraction is important for identifying the real and perceived trends in the phenomena		The widespread availability of computer programs allows for the automation of the data visualization. However, it is important to recognize the potential pitfalls that the user might inadvertently cause (e.g., data incorrectly formatted, missing data, etc.)
Analyzing Data	Analyzing requires the ability to identify potentially small- and large-scale patterns that might have arisen. Additionally, the recognition of the various components in the investigation that might influence the results are important.	The purpose of analyzing is finding and explaining the patterns	Explaining patterns requires using the data to make some generalizations beyond just the investigation	The ultimate goal of analyzing is identifying an algorithm that can be applied to explain the phenomena	Computer programs that allow for conducing inferential statistics (chi square, t-tests, etc.) can automate the process of analyzing the data.

From this crosswalk, we have developed an extensive set of questions that a teacher can use in the lesson planning process or that students can use while engaged in creating data, collecting data, manipulating data, or visualizing data. The questions are organized by data practice and then by computational thinking for each data practice. After we present the questions, we explain how this set of questions can be used.

What Questions Can We Ask to Support Computational Thinking in Data Practices?

Data Practice #1 - Creating: Generating data from tools or observation

Decomposition

1. *Break it down*: What do we already know about this problem? What data are we generating? What are the key types/sources of data that we might use to answer the essential question? How will we generate this data? What metrics can we use?
2. *Evaluate and choose*: Is the data primary or secondary? Which of the data sources will best help to answer our essential question?

Algorithm Building

1. *Generalize*: Which process will help us to effectively generate data to solve this problem? What are the rules or processes for creating data?
2. *Identify Conditions*: What conditions (e.g., if this, then that) need to be implemented to gather the data? What are the limitations or restrictions to the process of collecting data? Is there any background information we need to know to set this up?
3. *Put it in order*: What is the most efficient sequence to follow to generate this data?

Automation

1. *Identify tools*: What tools can we use/create to automate the generation of this data? How is the automation working to get the results? Does the automation help or hinder the data collection process?

Data Practice #2 - Collecting: Gathering and recording data

Pattern Recognition

1. *Identify patterns*: What are the intervals of the collected data? When does a data cycle end and when does it begin? Are we gathering enough data to see patterns or cycles in the phenomena?

Abstraction

1. *Identify essential elements*: How frequently do we need to collect data? Which data points matter to us? Do we have too much data?
2. *Identify noise*: Is there certain information that is less helpful when trying to figure out how best to gather data?

Algorithm (Checking)

1. *Generalize*: What are processes or rules that should be followed consistently and why? Are there ways to change the rules for data collection to improve our data collection process?
2. *Identify conditions*: What conditions or events happened during the data collection that might result in influencing the data beyond what you are testing? Under what conditions will you NOT collect data?

Automation

1. *Identify tools*: What tools could we use to collect the data more efficiently? More accurately?

Data Practice #3 - Manipulating: Sorting, filtering, cleaning, normalizing, and combining data sets

Decomposition

1. *Break it down*: Which data would you consider your independent variable? Dependent variable? What variables can we use to filter/sort the data? What are the different ways we might filter/sort the data?
2. *Evaluate and choose*: What steps must be undertaken to get the data into a format that will allow looking for patterns using the appropriate descriptive statistics and graphs?

Pattern Recognition

1. *Identify patterns*: In what ways does the data repeat? Are there bins or buckets we can use to group the data? Are there data missing or incomplete?

Abstraction

1. *Identify essential elements:* What information did you collect most necessary to answer your research question? Can you filter the data to only show the most important data?
2. *Remove noise*: Are there outliers that we can remove? How can we reduce noise in our data? What is your reasoning for removing them (just because they appear to be outliers doesn't justify removing them)?
3. *Create a model*: Can we simplify the data by sorting/filtering/cleaning it to more closely match our mental model?

Algorithm Building:

1. *Generalize*: What are the rules or processes for manipulating data?
2. *Identify conditions*: Under what circumstances do we use each form of data manipulation?

3. *Put it in order*: Is there an order to the steps in manipulating the data? What rules determine that order?

Automation

1. *Identify tools*: What tools can help us to filter, sort, or clean the data? Is there a way to more easily manipulate the data by automating all or some of the process?

Data Practice #4 - Visualizing: Communicating results with a representation such as a graph or chart

Decomposition:

1. *Break it down*: What are possible ways could we visualize this data?
2. *Evaluate and choose*: Which visualizations would help us to answer our essential question?

Pattern Recognition:

1. *Identify patterns*: What patterns can we see in the graphs/charts? How do the visualizations change as we add more data or manipulate the data?

Abstraction:

1. *Remove noise*: Can we adjust the type or scale of our visualization to focus on what we want to analyze? What is the clearest way of visualizing the data?
2. *Create a model*: How does this visualization match or contradict our mental model?

Automation:

1. *Identify tools*: Can we automatically update the visualizations as we add more data or manipulate the data?

Data Practice #5 - Analyzing: Extracting meaning from a data set for the purpose of drawing conclusions

Decomposition:

1. *Break it down*: What possible factors could help answer our question? Are there factors we haven't considered yet that we should?

Pattern Recognition:

1. *Identifying patterns*: Do the data look random? Are they regular? Are there increasing or decreasing patterns? What "shape" is the data?

Abstraction:

1. *Remove noise*: Is there noise affecting our data? What kind of effect is that noise having on our analysis?
2. *Create a model*: How can we model the relationships between variables? Is there a variable that affects the outcome more than the others?

Algorithm Building:

1. *Generalize*: Is there a formula we can use to describe the relationship between variables?

Automation:

1. *Identifying tools*: What tools can we use to analyze and model the data?

Using the Questions

The entire set of CT questions for data practices is not intended to be used all at once. Instead, teachers could use these questions in two ways. First, teachers could take a lesson that is already written and choose one or two data practices to focus on. For example, if the lesson calls for secondary data sets to be selected from the NOAA website, questions about creating data may not be as relevant, but questions about visualizing data may be more important. Alternatively, teachers could focus on one CT practice, such as decomposition, throughout the lesson and apply each decomposition question in the sequence of data practices from creating to visualization.

Students can benefit from having these questions embedded into lessons. For example, we asked students to conduct an investigation to find factors that influence the height of tides. In this investigation, we asked students to collect data on a NOAA website and to plot the height of tides for four locations. The data provided by NOAA can be downloaded by different time intervals. Many of the students downloaded the data using the same measure per day over the course of a year.

When students plotted the data, it appeared to be thick colored lines and students could not interpret the patterns (see Figure 1). We asked the students questions from the manipulating data section ("What information did you collect most necessary to answer your research question? Can you filter the data to only show the most important data?") Prompted by the

questions, students decided to use a smaller amount of data, every hour for a month, which was more appropriate for interpreting the phenomena.

With the manipulated data, students were able to see the patterns of the highs and lows in tides (see Figure 2). Further, students were able to discern that tide heights differed by location, which then led them to consider there is more than one variable affecting the height of tides.

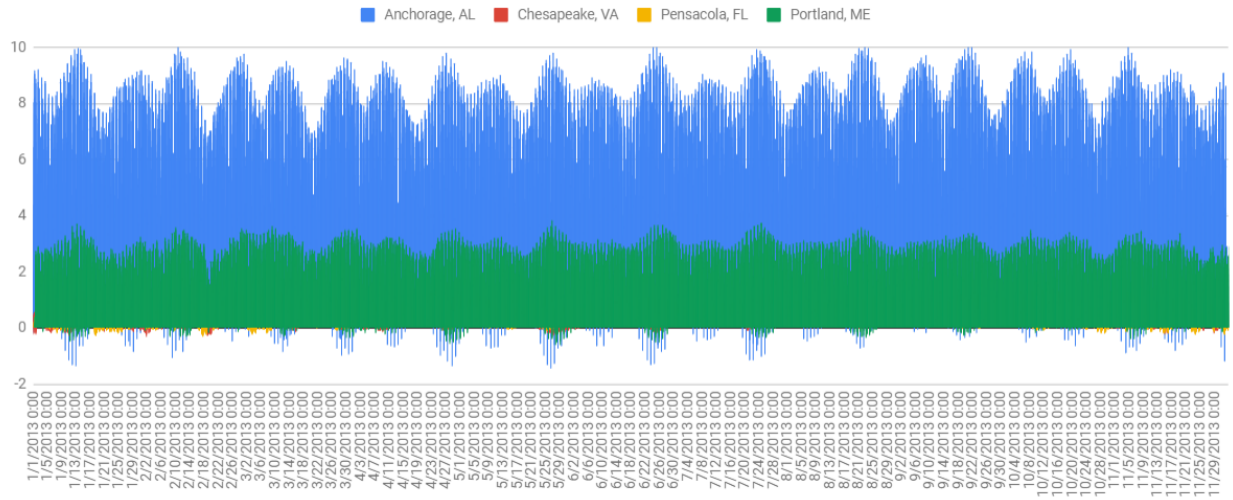

Figure 1. *Plot of tide height from NOAA website for four locations at the time interval once a week over one year. This plot demonstrates when too much data collection can prohibit pattern recognition.*

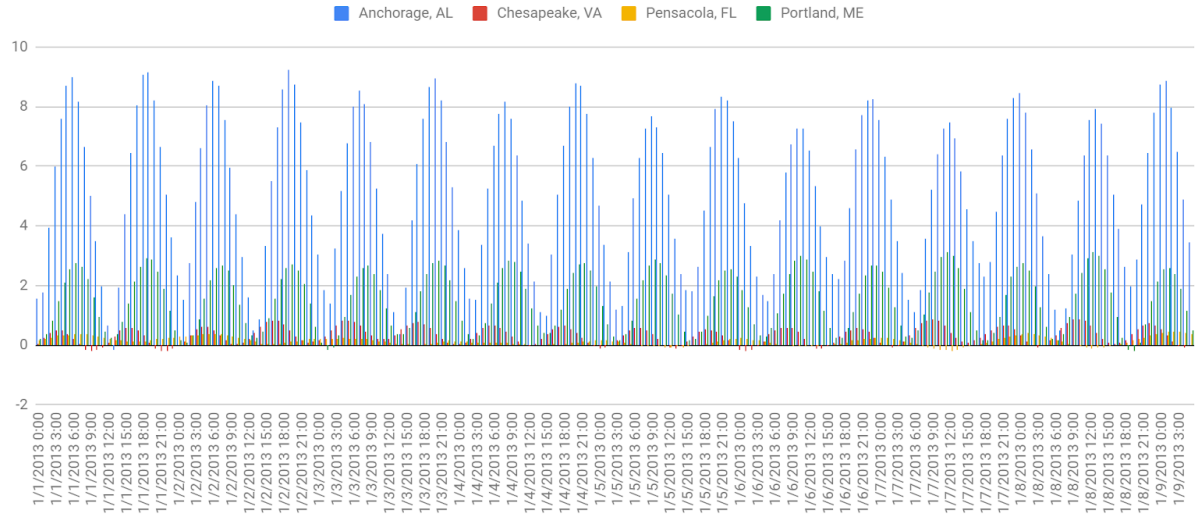

Figure 2. *Plot of tide height from NOAA website for four locations at time intervals of every three hours over a week. This plot demonstrates that the collection of data must be coherent with the patterns of the phenomena.*

By asking these CT questions about data practices, students can be more metacognitive about why they make the choices they make in their data practices. Understanding more about the assumptions made in data practices by using computational thinking could help teachers be more explicit in teaching about analysis of data and can help students think about why they are doing the data practices to make valid and rational decisions.

References

Augustine, N.R., et al. 2005. *Rising above the gathering storm: Energizing and employing America for a brighter economic future.* National Academies Press, Washington, DC.

Gilbert, J. K. (2004). Models and modelling: Routes to more authentic science education. *International Journal of Science and Mathematics Education, 2*(2), 115-130.

Foster, I. (2006). 2020 computing: a two-way street to science's future. *Nature, 440*(7083):419.

NGSS reference

Weintrop, D., et al. 2016. Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology* 25: 127–147. doi: 10.1007/s10956-015-9581-5.

Wing, J.M. 2006. Computational thinking. *Communications of the ACM* 49 (3): 33–35. doi: 10.1145/1118178.1118215.

Please provide author info (title, affiliation):

Erin Peters Burton, Donna R. and David E. Sterling Endowed Professor in Science Education, George Mason University;

Peter Rich, Peter J. Rich, Associate Professor, Brigham Young University;

Timothy Cleary, Associate Professor, Rutgers University;

Stephen Burton, Science Outreach Teacher, Loudoun County Public Schools;

Anastasia Kitsantas, Professor, George Mason University;

Garrett Egan, Graduate Student, Brigham Young University;

and Jordan Ellsworth, Graduate Student, Brigham Young University