Concepts, practices, and perspectives for developing computational data literacy: Insights from workshops with a new data programming system

Ruijia Cheng
Aayushi Dangol
Frances Marie Tabio Ello
rcheng6@uw.edu
adango@uw.edu
fello@uw.edu
University of Washington
Seattle, Washington, USA

Lingyu Wang
lyw@unc.edu
University of North Carolina at Chapel Hill
Chapel Hill, North Carolina, USA

Sayamindu Dasgupta[*]
sdg1@uw.edu
University of Washington
Seattle, Washington, USA

ABSTRACT

In this paper, we present a new visual block-based programming system designed for children to process, analyze, and visualize data. We introduce the system and describe how it was used during a series of 7 workshops with 27 children. During the workshops, children played the role of investigators and followed a storyline as part of the system to conduct data analyses to help the story's protagonist locate a missing family member. We present our findings as a framework of computational data literacy that builds on the dimensions of Computational Thinking proposed by Brennan and Resnick [8], with a focus on aspects that are specific to using programming for data processing, analysis, and visualization. We conclude with a series of recommendations for future designers of systems to support the development of computational data literacy.

CCS CONCEPTS

• **Social and professional topics → Computing literacy**; • **Human-centered computing** → *Visualization toolkits.*

KEYWORDS

block-based programming, data literacy, visualization

ACM Reference Format:
Ruijia Cheng, Aayushi Dangol, Frances Marie Tabio Ello, Lingyu Wang, and Sayamindu Dasgupta. 2023. Concepts, practices, and perspectives for developing computational data literacy: Insights from workshops with a new data programming system. In *Interaction Design and Children (IDC '23), June 19–23, 2023, Chicago, IL, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3585088.3589364

1 INTRODUCTION

Being able to make sense of our world through data has become an increasingly important ability. Significant interest has emerged in approaches that could support the development of this ability in middle- and high-school-aged young people [30, 42]. Often referred to as *data literacy*, a key requirement of this skill is to become familiar with practices around data, such as reading data, working with data, analyzing data, and arguing with data [18]. A subset of these practices is often realized through computational means, commonly by writing computer programs. For example, analyzing a large dataset typically involves writing a program to filter and reshape the data, which can then be used as input for another bespoke program that generates a visualization. Compared to pre-programmed tools (e.g., Microsoft Excel) for data processing, programming with data offers a wider array of possibilities (e.g., new forms of data visualization) [12, 24]. Although programming with data is a practice seen primarily among professionals (e.g., professional data analysts) [27, 29], recent discussions of democratizing data science call for everyone, including young people, to learn computational skills that support asking and answering questions with data [24].

This paper showcases our attempt to bring the power of programming with data to children. We focus on the computational aspects of data literacy, which we refer to using the term *computational data literacy* [56]. We first introduce *Dataland*, a visual block-based programming system designed for young people that focuses on data analysis and visualizations situated in story contexts. We describe the system and then report on workshops where children used the programming system and played the role of an investigator to conduct a series of data analyses in order to help the story's protagonist locate a missing family member. We build on Brennan and Resnick's [8] framework of Computational Thinking and present our findings as a framework for studying computational data literacy. We conclude with a discussion of our findings, including recommendations for future designers of systems to support the development of computational data literacy. Our contributions are as follows.

- A visual block-based programming system to analyze and visualize data in the context of narrative storylines.

[*] Part of this work was done when Dasgupta was at the University of North Carolina at Chapel Hill.

IDC '23, June 19–23, 2023, Chicago, IL, USA

ACM ISBN 979-8-4007-0131-3/23/06.
https://doi.org/10.1145/3585088.3589364

- Empirical findings from workshops where young people used the system.
- An extension of an existing framework that presents computational data literacy in terms of three distinct dimensions—concepts, practices, and perspectives.
- Recommendations for future designers of systems that support the development of computational data literacy.

2 RELATED WORK

2.1 Computational data literacy: youth data literacy and computational thinking

In today's data-driven world, the ability to understand and use data has become crucial not only for professional data scientists but also for everyday people [16, 24]. Argued by many to be a new form of literacy, *data literacy* has been brought up as an essential skill that young people should acquire [30]. Existing scholarly work has explored and proposed different dimensions of data literacy. Many agree that data literacy for young people includes the ability to read data (e.g., understanding what a dataset represents), work with data (e.g., cleaning a dataset), analyze data (e.g., filtering data), and argue with data (e.g., visualizing data to support a claim) [18, 47]. Data science education research also emphasizes the ability to understand the context and human factors that affect data collection and analysis, as well as to aggregate, visualize, and make inferences with data [42]. Additionally, increasing attention has been paid to the critical aspects of data literacy, (e.g., the ability to decipher how data is created) [17, 20, 51], as well as the skills to access and interpret data through the lens of community and social impact [19, 23, 35].

Certain aspects of data literacy are associated with the learner's ability to program with data. For example, one can write programs to efficiently and reliably sort, filter, clean, join, and make meaningful visualizations of large and complex datasets [50]. In this paper, we focus on programming with data, and in order to differentiate this from broader data literacy that may not involve programming, drawing from Yalcinkaya et al. [56], we call the ability to work with data through programming *computational data literacy.* Apart from data literacy, computational data literacy also connects to the broader notion of Computational Thinking (CT), the ability to solve problems "by drawing on the concepts fundamental to computer science. [53, p. 33]" CT has been studied and theorized extensively. For example, Brennan and Resnick [8] offer a well-known framework for CT that decomposes it into a series of *concepts* such as sequences, loops, and conditionals; *practices* such as testing, debugging, and abstracting; and *perspectives* of what CT enables, such as expressing oneself, connecting with others, and questioning technology. In more recent work that is closer to our topic, Basu et al. [2] identifies a set of "focal knowledge, skills, and abilities" for assessing the concept of "Data and Analysis" as defined by another CT framework [37], and Berikan and Özdemir [4] highlighted that "problem solving with datasets" as a key implementation of CT. Our work builds on these prior works and contributes a complementary series of concepts, practices, and perspectives that are unique to computational data literacy.

2.2 Designing systems and scaffolds to foster computational data literacy in children

In recent years, a number of systems and scaffolds that engage young people in analyzing and visualizing data have emerged. For example, the Quantified Self movement—a global trend where sensors on mobile and wearable technologies are used to collect data on one's own everyday activities—has been seen as a rich environment for learning with and about data [31]. The Common Online Data Analysis Platform (CODAP) [21] system allows students of grades 6–12 to access and explore data from a wide variety of sources. Dasgupta and Hill [12] designed Scratch Community Blocks, a system built on top of the Scratch programming environment and the online community [40] that allowed members of the Scratch community to programmatically access, analyze, and visualize their own activities in the community. In addition to inventing novel systems, researchers have also been developing toolkits, curricula, and community activities that support youth data literacy. Examples include real-world datasets prepared for a variety of educational contexts [1], as well as classroom and outreach activities developed to engage students and community members in recognizing bias and complexity in data and producing data artifacts for social impact [6, 13, 14, 18].

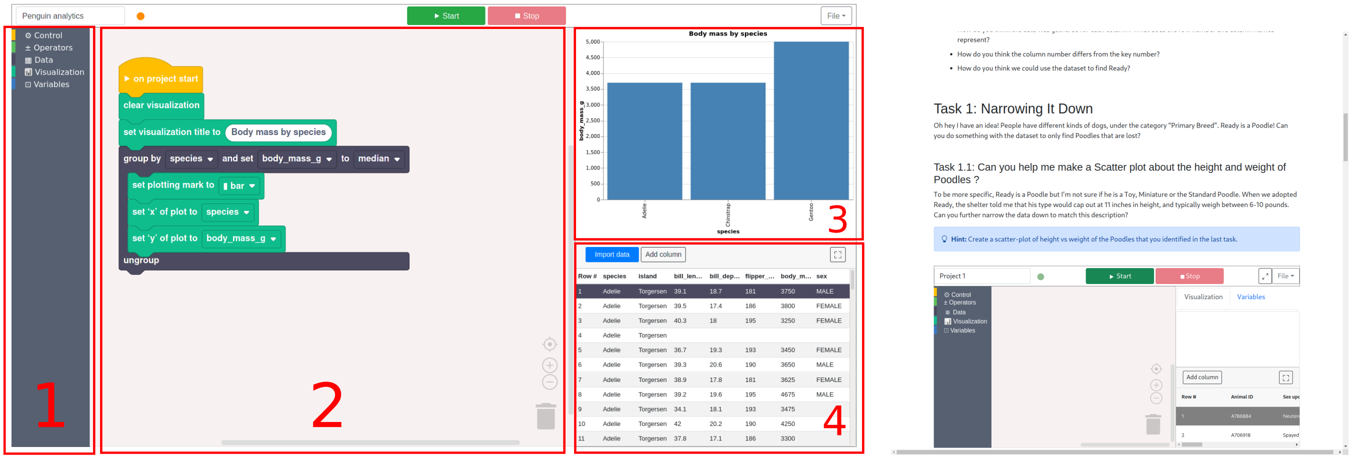
Despite the pluralism in data literacy tools, few systems specifically support computational data literacy in youth or directly support young people with limited programming experience to effectively program with data. While Scratch Community Blocks [12] allowed children to program with data using visual blocks, the data was limited to those of their activities in Scratch and the programming grammar and conventions were constrained to that of Scratch. Building on prior work and to explore the design space with fewer constraints, we present our design that supports youth computational data literacy. We draw design guidance from principles of Constructionism, a framework that has been widely used to design systems that support the learning of mathematical knowledge and programming skills [26, 36]. Following the Constructionist principles of "low floors" for novice entry, "high ceilings" for enabling advanced possibilities, and "wide walls" for exploration [39], we designed a visual block-based programming system called *Dataland* that offers several built-in data query and visualization programming primitives. Inspired by prior work that recognizes the importance of situating data analysis within a broader narrative context [10, 31, 38, 43], we implement data stories in *Dataland* that position users in a personally engaging narrative.

3 SYSTEM DESIGN

Dataland is presented through a "storyline," (Figure 1b) a web-based interface that tells a data story (described in §3.2). Throughout the storyline are several instances of code editors (shown in Figure 1a), where users can work on block-based programming projects. In this section, we describe the code editor and the storyline interface. Readers can access the code editor on our project website, https://learning-with-data.github.io/.

3.1 Language design: Vocabulary and grammar

The *Dataland* code editor contains four main components: 1) a block palette containing all the programming blocks; 2) a coding



```
▶ on project start
clear visualization
set visualization title to (Body mass by species)
group by [species ▾] and set [body_mass_g ▾] to [median ▾]
    set plotting mark to [▮ bar ▾]
    set ‘x’ of plot to [species ▾]
    set ‘y’ of plot to [body_mass_g ▾]
ungroup
```

Row #	species	island	bill_len...	bill_dep...	flipper_...	body_m...	sex
1	Adelie	Torgersen	39.1	18.7	181	3750	MALE
2	Adelie	Torgersen	39.5	17.4	186	3800	FEMALE
3	Adelie	Torgersen	40.3	18	195	3250	FEMALE
4	Adelie	Torgersen					
5	Adelie	Torgersen	36.7	19.3	193	3450	FEMALE
6	Adelie	Torgersen	39.3	20.6	190	3650	MALE
7	Adelie	Torgersen	38.9	17.8	181	3625	FEMALE
8	Adelie	Torgersen	39.2	19.6	195	4675	MALE
9	Adelie	Torgersen	34.1	18.1	193	3475	
10	Adelie	Torgersen	42	20.2	190	4250	
11	Adelie	Torgersen	37.8	17.1	186	3300	

represent?

- How do you think the column number differs from the key number?
- How do you think we could use the dataset to find Ready?

Task 1: Narrowing It Down

Oh hey I have an idea! People have different kinds of dogs, under the category “Primary Breed”. Ready is a Poodle! Can you do something with the dataset to only find Poodles that are lost?

Task 1.1: Can you help me make a Scatter plot about the height and weight of Poodles ?

To be more specific, Ready is a Poodle but I’m not sure if he is a Toy, Miniature or the Standard Poodle. When we adopted Ready, the shelter told me that his type would cap out at 11 inches in height, and typically weigh between 6-10 pounds. Can you further narrow the data down to match this description?

Hint: Create a scatter-plot of height vs weight of the Poodles that you identified in the last task.

Row #	Animal ID	Sex upc
1	A786884	Neuter
2	A706918	Spayed

(a) The *Dataland* code editor user interface with the un-expanded block palette ①, the coding area ②, the data visualization area ③, and the data table ④

(b) The storyline interface showing the editor embedded within the narrative text.

Figure 1: The *Dataland* code editor and story interface.

area where users can program by dragging and dropping blocks; 3) a panel that shows visualization created by code and the dataset in the data table; and 4) a view of a data table that can be pre-populated or imported from a CSV file and edited (via code or the “Add column” button) by a user. The visualization system of the editor is implemented through *Vega-Lite* [44] and Leaflet.js [48], and the block-based editor is implemented through the Blockly framework [22]. In Figure 1a and the rest of this section, we use the Palmer Penguins dataset [25] for illustrative purposes.

In *Dataland*, the programming blocks are the primitives that form the vocabulary of the programming language. In a way similar to how text is constructed with the vocabulary of a natural language, children who use *Dataland* will compose these primitives to create programs. The design of the *Dataland* programming blocks follows principles in constructionist learning theories, where the choice of these building blocks “determines, to a large extent, what ideas users can explore with the kit—and what ideas remain hidden from view” [41, p. 119]. We adopted much of the design of basic programming blocks (e.g., mathematical operations, conditions, loop) from the design of the Scratch programming blocks [34, 40] and added new blocks that are specific for data analysis. In this section, we focus on aspects of the language design that are unique to *Dataland*. Along with individual blocks (which represent the vocabulary of the language), we also provide an overview of the grammar—the rules that govern how the blocks can be composed. Drawing from Resnick and Silverman [41], especially the principle of “inventing things that you would want to use yourself,” the starting point of these design features are our experiences of working and educating with, as well as designing existing data programming systems and platforms.

3.1.1 *Navigating data table and accessing data.*

A fundamental requirement for any data programming system is to allow for accessing or reading data. Data is imported into *Dataland* through a user-interface gesture (a button click) or can be predefined in a storyline. After that, users can interact with imported data through programming. Following the design principles of “making execution visible” and “making data concrete” from Scratch [34], during *Dataland*’s program runtime, a specific row is always in a selected state in the data table, and any changes in row selection and cell values are reflected in real time.

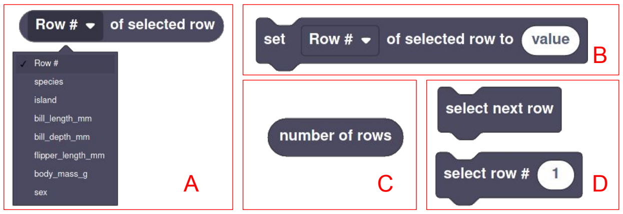
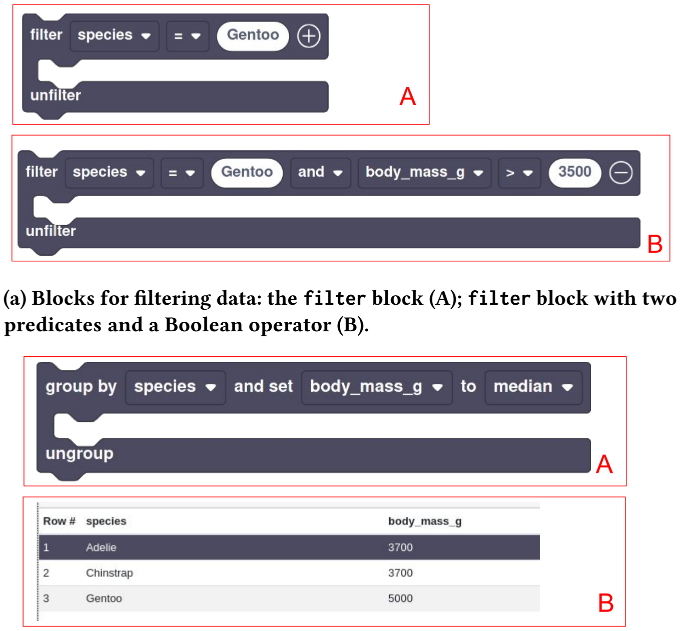


Figure 2: Blocks for accessing data: the Data access block with column drop-down (A); Block to set value in a row (B); Reporter block that returns total number of rows (C); Blocks for selecting data rows (D).

For programmatic access, we designed a set of unique data blocks (Figure 2). To change row selection, users will need the `select next row` or `select row # _` block (Figure 2 D). To access specific value from a row, the `_↓ of selected row` “reporter” block has a drop-down menu (indicated with the ↓ symbol) with a list of column names (Figure 2 A). This reported value can then act as an argument or input for another block (e.g., a mathematical operation block). Finally, a `number of rows` reporter block allows users to set up loops (Figure 2 C) to traverse the entire set of rows. In addition to access, users can also set specific values within a row with the `set _↓ of selected row to _` block (Figure 2 B). If a new column is needed (e.g., a derivative column), it can be added via a button click on the user interface.

3.1.2 *Filtering.*

Filtering data in *Dataland* is made possible by the `filter` block that we designed. Filtering data is a common data



Row #	species	body_mass_g
1	Adelie	3700
2	Chinstrap	3700
3	Gentoo	5000

(a) Blocks for filtering data: the `filter` block (A); `filter` block with two predicates and a Boolean operator (B).

(b) Block for aggregating data and temporary results on the data table view: the `group by` block (A); The state of the data table view when the `group by` block is running (B).

Figure 3: Filter and aggregation blocks.

analysis task that, at the conceptual level, builds on logical true-false predicates combined with Boolean operators, resulting in data transformations (e.g., a subset of the original table being returned). We designed the `filter` block as a c-shaped block, or "c-block", which is usually used to indicate contexts where temporary operations on the filtered data might take place. We further added labels "filter ..." and "unfilter“ at the top and bottom of the block to suggest that the original dataset will be restored after the operations on the filtered data are complete (Figure 3a A). In addition, since it is common in data analysis tasks to specify variable-length conditions for filtering, combined with Boolean operators (e.g., *species* == *Gentoo AND body_mass* > 3500), our design allows multiple predicates in the `filter` block (Figure 3a B). With this design, users can start with a single condition and later add another condition by clicking on a ⊕ button that would expand the block to include an additional condition, and combine the two with a Boolean operator. For simplicity, in our prototype, we have restricted the block to two conditions at most. Filter c-blocks can be nested if required, offering an alternative way of specifying additional predicates for filtering.

3.1.3 Aggregation. Aggregation, or grouping, is another common task associated with data analysis. We follow a model similar to filtering as described above. A c-block (Figure 3b A) allows the *Dataland* user to specify the variable to group by and also the variable to apply the aggregate function on. Aggregate functions, including `count`, `count unique`, `maximum`, `minimum`, `mean`, `median`, `mode`, and `sum`, can be chosen from the drop-down menu. Inside the c-block, code blocks can access the group and the corresponding aggregation result. For example, Figure 1a shows how the `group by` block may be used to create a visualization of grouped data. As with the `filter` block, the visual representation of the data table is updated to show the grouped data when the `group by` block is executing (Figure 3b B), and an "ungroup" label is added to the bottom of the c-block to indicate that the operation is temporary.

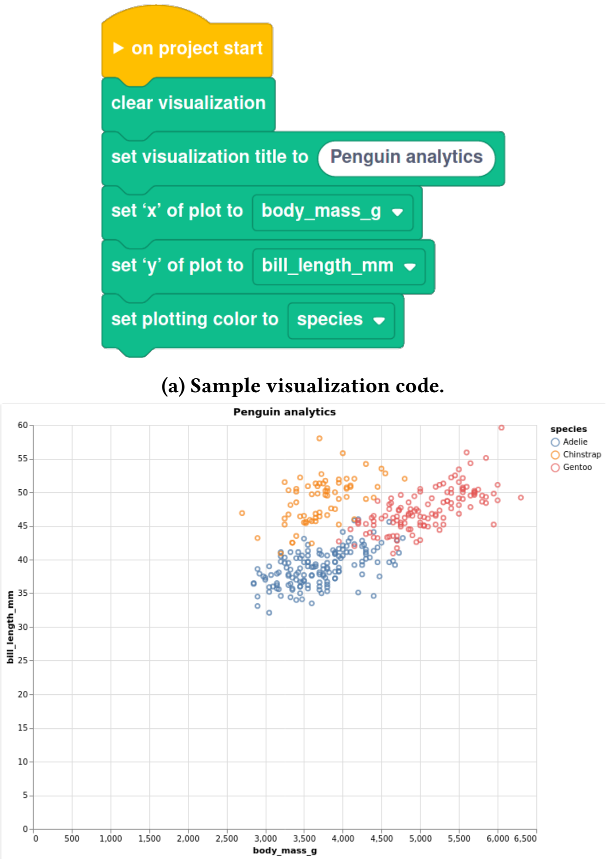
3.1.4 Visualization. Data visualization is one of the major ways for data analysts to communicate their findings and conclusions, and it is also a crucial tool for exploring datasets and finding potential relations among variables or data columns. *Dataland* supports a set of "canonical plots": the standard and commonly used types of Cartesian plots that include scatter plots, line plots, and bar plots. To design visualization code blocks, we followed Wilkinson's grammar of graphics [52] and *Vega-Lite* [44] specifications closely (Figure 4) to have atomic functions, with fewer blanks for users to fill in. We have also simplified Wilkinson's plotting process by hiding and automating certain steps [52, p. 39]. Here again, we are working with two of the design principles suggested by Resnick and Silverman: that we should keep the floor low for novices, while finding the simplest way to do the most complex things [41, p. 119]. Figure 4a demonstrates how columns in a dataset (attributes of the data) are coded for visualization, and Figure 4b shows the result.

In addition to blocks for canonical data visualizations, *Dataland* also supports plotting on geographical maps. The process is similar, but partly due to technological constraints and partly to keep the number of blocks being presented to learners minimal, blocks for map-based visualizations are available in a separate version of the editor.

3.2 Storyline

Computational projects by young people (e.g., in Scratch) are often in the genre of games, animations, interactive stories, etc. Works in these genres usually stand alone in our culture (e.g., we come across games as standalone cultural objects), whereas computational data analysis projects usually are encountered within a broader context (e.g., as a part of a newspaper report). This prompted us to consider the importance of embedding data analysis projects within story-like broader contexts. As a result, a computational notebook-inspired [28] story-based interface (referred as "storyline") was created for users of *Dataland* to research and analyze data to find answers (Figure 1b). Following the storyline, we expected that learners will explore data, gain insights by filtering and cleaning data, produce data visualizations, and interpret their findings—all essential skills for building data literacy.

We developed 3 storylines, which we refer to as "Penguin A", "Penguin B", and "Poodle" in this paper. We developed "Penguin A" as the initial story for our system, and "Penguin B" is a close derivative of "Penguin A" that we developed based on feedback and observations from our first series of workshops. We developed "Poodle" based on our insights from the earlier workshop series and also made it locally relevant to our research setting, for example, by incorporating references to neighborhoods and landmarks in the city where we conducted our workshops.



```
▶ on project start
clear visualization
set visualization title to (Penguin analytics)
set ‘x’ of plot to [body_mass_g ▾]
set ‘y’ of plot to [bill_length_mm ▾]
set plotting color to [species ▾]
```

(a) Sample visualization code.

(b) Result of running sample code.

Figure 4: Visualizing data with *Dataland*

Penguin A and B were developed around the tale of a traveling penguin who is searching for their missing cousin after a snowstorm, while Poodle centers on a young boy and his missing dog. For both stories, the learner using *Dataland* would take on the role of a researcher assisting in the search. Penguin A and B are based on the Palmer Islands penguin dataset [25], and Poodle is based on a public-domain children’s storybook by Mabel Stryker [45] and uses data sourced from a combination of Seattle and Austin open data portals. For all the stories, we had to add some synthetic data (e.g., new columns) to the original datasets to incorporate the computational data literacy concepts that we aimed to cover.

Each story was structured as a set of interconnected puzzles, with each puzzle being a data analysis task that could be done with a *Dataland* program. For designing the tasks, we drew inspiration from *Sahaj Path*—a primer for the Bengali language, written by Rabindranath Tagore [46]. Each short story or poem in *Sahaj Path* emphasizes a specific alphabet, and we followed this approach by designing each task to focus on one or two data literacy concepts (e.g., filtering). The text of our storylines can be found in the supplemental material of this paper.

4 USER STUDIES: *DATALAND* WORKSHOPS

4.1 Workshop and participants

We ran a total of 7 research workshops in 2022 with 27 participants (referred to as P1 to P27) in total. There were 9, 7, 1, 4, 2, 3, 4 participants who participated in the workshops on May 14th, May 15th, June 4th, June 5th, November 9th, November 12th, and November 13th, respectively. Most of the participants only attended one workshop, with the exception of P9, P3, and P12. P9 attended both of the May workshops that used the Penguin A storyline, and P3 and P12 who each attended a May workshop and a June workshop. We used the Penguin A storyline (see §3.2) in the first 2 workshops in May (with P1, P3, P5 - P17), the Penguin B storyline in 2 workshops in June (with P2 - P4, P12, and P18), and the Poodle storyline in the remaining 3 workshops in November (with P19 - P27).

All the workshops were hosted on a weekend morning or a weekday evening at our institute and lasted around 3.5 hours. During the workshop, participants were instructed to interact with the *Dataland* system to follow a storyline to conduct data analysis. Researchers in our team facilitated the workshops, taking field notes, and asking questions to participants. At the end of each workshop, participants were given the option to participate in an interview, where they reflected on their experience using *Dataland* and working with data. Details on the profile of our participants, how we selected the participants, and how we engaged with the participants in the workshops can be found in §8.

4.2 Data and Analysis

The field notes and interviews were transcribed by the workshop facilitators. A total of 27 entries of field notes and 24 interviews were collected from the 7 workshops. The duration of the interviews ranged from 5 to 25 minutes, with an average duration of 11 1/2 minutes. We collected a total of 90 pages of field notes.

We followed a thematic analysis procedure [7] to analyze field notes and interview data. Three researchers who had facilitated workshops first independently annotated lines of interview transcripts and field notes with notes for possible themes. This round of coding was guided by the structure of the Computational Thinking framework by Brennan and Resnick [8] and focused on constructing themes that fell in the categories of “concepts”, “practices”, and “perspectives”. The research team then discussed the codes, identified common themes, reached a consensus on the codes, and collaboratively constructed a codebook. The researchers then reconducted two additional rounds of coding to iteratively merge and synthesize a final set of themes, which are presented in the following section.

5 FINDINGS

In this section we report our findings as a framework for studying and developing computational data literacy, following the categorical structure of Computational Thinking proposed by Brennan and Resnick [8]—concepts, practices, and perspectives. At a fundamental level, participants in our studies were writing computer programs in a language that shared some characteristics with Scratch, and hence engaged with at least a subset of Brennan and Resnick’s concepts, practices, and perspectives. However, in this section, we focus on aspects of these dimensions that go beyond what has already been described by Brennan and Resnick and are more specific to computational data literacy. Building on evidence from our *Dataland* workshops, our framework describes the new **concepts (CO)**, **practices (PR)**, and **perspectives (PE)** that learners can learn through programming with data, and we list and define them as follows.

5.1 Concepts

In this section, we present several of the key *concepts* of computational data literacy that we observed from the *Dataland* workshops, including **CO1:** Filtering, **CO2:** Aggregation, **CO3:** Variables, **CO4:** Statistical Concepts, and **CO5:** Visualizations. In particular, we focus on how participants achieved understanding of these concepts and the underlying mechanisms through engaging with *Dataland* in our workshops.

5.1.1 CO1: Filtering. One of the most important concepts that participants learned about using *Dataland* is *filtering*—narrowing down a given dataset based on one or more conditions. Following a given data story, participants applied the `filter` block on the dataset to make visualizations and solve problems. In this process, participants were able to reach a practical understanding of how filtering data works.

Many aspects of the design of *Dataland* helped participants figure out how filtering data works. The c-shaped design of the `filter` block offered visual suggestions that data would be filtered "inside" the block and would be "given back" once the program finished running inside the block:

> "At the bottom, as in was like it's shaped like a C, right? And everything that is inside the C happens when it applies. And everything that is on the bottom of the C happens when it doesn't apply." (P19)

Additionally, being able to see in real time how the execution of the program would impact the data table helped participants understand filtering. For example, P21 made a program that contained a `filter` block. In order to figure out how filtering worked, P21 looked at the data table while the program was being executed. They[1] observed that once the program entered the filter, the data table will switch to the filtered version that contained fewer rows, and switched back once the execution finished. They thus realized that the code inside a `filter` block ran on the filtered dataset.

However, a challenging aspect of filtering for participants was understanding that filtering would not permanently change the original dataset. Initially, many participants expressed concerns and were cautious about getting their datasets altered by the `filter` block, despite multiple design choices to alleviate that impression. We believe that additional instruction and explanation would be needed to explain that the filter operation is temporary.

Most participants were also able to use the ⊕ button or stack `filter` blocks together to add additional conditions when filtering data. They were able to understand the Boolean operators between conditions. For example, P25 explained the relationship between two conditions in a filter: "'and' is considering both parts, and 'or' is either parts."

5.1.2 CO2: Aggregation. Another important and complex concept that *Dataland* offers to teach was *aggregation*—grouping data based on certain categorical attributes and then performing operations on the groups. At first, many participants did not understand what the `group by` block was and were confused about what the output of aggregation would look like. For example, based on the name of the block, some participants initially guessed that it would help them sort data into groups: "I thought it would like put things into categories. But I don't think it does that." (P19) Through playing with the `group by` block and plotting data, participants were able to gradually realize key underlying concepts. Participants explained aggregation in their own words:

> "I would say that the first [drop-down in the `group by` block] is like, what kind of groups you're separating it into, and then the last one is how you want it done—do you want it just the maximums, the means, the sums, [or] the counts? " (P14)
> "`group by` is [to] find the types in a variable and `set _ to _` is to do the math on a single column." (P24)

Participants later reflected on what made aggregation confusing. Many pointed to the design of the `group by` block. For example, P14 thought "there were too many options" in the block. P24 further pointed out that the first two entries in the block both being drop-down menus could be a source of confusion, since users might not understand the different roles played by the columns selected from the drop-down menus. Participants also brainstormed about how to improve the `group by` block to make it more straightforward. For example, P3 thought about changing the grammar: "instead of saying like, `group by _, set _ to _`, it can be like, `find` the minimum of this particular category of each, and then whichever you're grouping by."

5.1.3 CO3: Variables. Participants also learned to use the concept of variable to store results and facilitate operations on data in our workshop[2]. One salient example was the storyline tasks that ask users to count the number of items that fit certain conditions. To do this task, participants needed to understand how to create a variable and use the `set variable` block to assign a value to it. Most participants were able to successfully learn to count using variables and understand the concept of variables despite initial challenges.

For example, P12 made a program that used the `filter` block, the `number of rows` block, and variables to count the number of penguins that fit certain conditions in the Penguin dataset. They explained the process of using variables to count data as: "the set block sets the variable that I made to whatever numbers it is, in this case is the number of rows of Adelie [penguins] and those smaller than 4500 grams." In another instance, P4 made a loop to solve a similar counting problem and in this process, they learned about variable initialization and increment: they first created a variable called `Adelie_count`, set it to 0 using the `set variable` block, then used the `repeat` block to create a loop. P4 explained this

[1] As we did not collect information on the participants' pronouns, we use the gender-neutral pronoun "they/them" while reporting the findings.

[2] The Brennan and Resnick framework covers variables—in this paper, we highlight a few data literacy specific aspects of the concept of variables.

program that they created as "each time it increases, I will increase `Adelie_count` by one."

In cases like these, participants not only learned about assigning values to variables, but also understood the nature of variables as a place to store values that could dynamically change. An important aspect to note here is that the `repeat`-based approach adopted by P4 does use the `number of rows` block as input for `repeat`, so P4 was certainly aware of the existence of that particular block. Even then, P4 added what was essentially redundant code to count the rows by incrementing the variable. This suggests that, for learners to understand the idea of operating on entire datasets, (described in §5.2.1) instructors may need to provide more active support.

5.1.4 CO4: Statistical concepts. Participants learned to apply statistical concepts when working with data. Although *Dataland* did not explicitly teach any statistical concepts to its users, some statistical concepts were embedded in the activities. Participants were exposed to statistical concepts as they went through the storyline and worked on the problems. In this process, they were able to empirically construct their own understanding of particular statistical concepts, such as types (categorical, numerical, etc.) and statistical operations (mean, median, etc.).

For example, the column containing age data in the Poodle dataset was not numerical, with values such as "7 years", "8 years". After a few attempts to use the `filter` block to filter ages that were lower than 7, P22 realized that the data in the age column were not numerical whereas they were putting a number into the `filter` block: "The data presents the age as a number and a word, so you can't really compare it [with a number]." (P22)

Another example is when P25 was calculating the mean weight of each breed of Poodle. P25 was initially confused about how to approach this question, and after discussion with a workshop facilitator, P25 realized that the mean equaled to the average, which was the sum of the data divided by the total number of data points. P25 then proceeded to implement a program that added all the weights by looping through all rows, stored the value in a variable called `Total`, and then divided `Total` by the number of rows.

5.1.5 CO5. Visualizations. Participants learned about creating and reading different visualizations in *Dataland* workshops. Most had never made visualizations before the workshop and therefore found the task unfamiliar and challenging at first. Nevertheless, towards the end, they were able to successfully create plots using the visualization blocks.

For example, P3 thought it was challenging to "figure out how the visualization works and how to actually visualize the stuff I want... what I'm supposed to set what to create the specific visualizations." In many cases, workshop facilitators stepped in and discussed the different plot types (in the case of our workshops, the bar plot and the scatter plot) and parameters. Participants then experimented with the visualization blocks before successfully making the plots.

We also observed participants experimenting with visualization blocks in different orders and combinations and learning about mechanisms in visualizations in the process. For instance, when trying to plot filtered data, P12 played with the order of the `set x to _` block, `set y to _` block, and a `filter` block. When asked why they eventually decided to place the `set x to _` block and `set y to _` inside the `filter` block, P12 explained that "because [otherwise] it will be plotted before the filter."

With the designed activities in the storyline and interactive visualization panels, participants were able to read and make meaningful interpretations on visualizations. For instance, working with the Penguin B storyline, we observed that P2 first plotted the flipper length and weight of different penguin species in different colors, then filtered for the Adelie and Chinstrap species and plotted again. P2 explained their thought process: "Because the cousin is small, and the Adelie and Chinstrap are the smaller ones. I plotted these two only to see [more clearly] which one is smaller."

5.2 Practices

In this section, we outline several key *practices* of computational data literacy that we identified from our data, including **PR1:** Conducting operations on entire datasets; **PR2:** Processing data and using the results; **PR3:** Tinkering with data and code; **PR4:** Cross-checking data and debugging with data; and **PR5:** Iterating on data presentation. We describe the various practices that participants demonstrated while working with data in *Dataland.*

5.2.1 PR1: Conducting operations on entire datasets. One important practice that participants developed using *Dataland* was to operate on the entire datasets all at once. While many participants were familiar with iteration and loops from previous experience of programming languages like Scratch, most of the participants had never been exposed to dataset-wide operations and would initially approach problems with loop programs.

The introduction of dataset-level operations was eye-opening for many participants, as said by P21, "do I not have to read through each row to plot the data [when filtering]?" P3 appreciated the `filter` block to allow them working with data in batches, so that they could increase the efficiency of their code: "I'm trying to figure out which piece I needed to use, how it could most effective, how I could deal with a specific process in the least amount of blocks." Participants also reflected on when to use batch operations versus when to iterate through data. For example, P24 compared `filter` and `if-else`: "Filter is getting all data and filter out in the dataset, if else is to get the specific thing...if else is, for those in the filter, you do the specifics and assign values." Similarly, P19 summarized filter as an operation that "takes your dataset:" "If-else is if this is true than this, but `filter` is like, for all that this is true, do this."

5.2.2 PR2: Processing data and using the results. Participants showed the practice of processing data, for example, adding new data to and manipulating row and columns in the data table. Rather than using pre-processed data for visualizations, *Dataland* allows users to develop the practice of processing data first. In our storylines, participants were guided to process the data in ways that were helpful to solve the problems that they were presented with. For example, P25 explained what they did to add a column for Poodle types: "I looped through all the Poodles, see if it fits any of the conditions, then set the value, then go to the next one." Once finishing processing data, it was common for participants to save the updated data table and upload it to an empty editor to analyze and plot the new data.

In addition to following the storylines, some participants also spontaneously added columns and data values as an intermediate step in solving the problems. For example, P24 created columns to store the total number of Poodles as a step towards creating a bar plot; P5 created columns to store the temperature of each year when working on a prediction problem. P26 suggested a new feature in *Dataland* that could "filter out the blanks" to clean up the missing values as part of data processing.

5.2.3 PR3: Tinkering with data and code. Participants demonstrated the practice of tinkering with data, that is, working with data in trial and error and constructing knowledge about data and operations in the process. Like any other visual block-based programming system, *Dataland* allows users to easily tinker with their programs—however, in this case, the tinkering was with *both data and code.* During programming, workshop participants tended to check the data table, store outputs in variables, or make plots to see the results as part of the trial-and-error process. For example, when using the `group by` block, P24 was not sure what `set _ to unique count` was. They used this block on a Poodle type and got 30 as a result. They were initially surprised because, from a previous task, they already found the count of the same Poodle type to be 87. Then they realized that "multiple dogs can have the same weight and it is counting thing[s] with the same value as 1," thus understanding how `group by` worked through tinkering.

5.2.4 PR4: Cross-checking and debugging with data. We observed many participants engage in the practice of cross-checking data and debugging with it. With the data table right by the side of the editing window, participants were able to refer to the data table after getting results or plots from the analysis to verify the results and spot any bugs. For instance, P24 created and ran a program to fill data into a new column based on certain conditions. When checking the newly added column, they found that only the first three rows were filled in. They initially tried to guess the reason as "the `if else` block does not take doubles." But when checking the dataset again on the three rows that were filled, they found that some of them did have decimals. After the help of a workshop facilitator, they finally found out that there was a missing `select next` block and their loop was "stuck at the first round."

5.2.5 PR5: Iterating on data presentation. We observed that participants were able to iteratively fine-tune the presentation of data to effectively communicate ideas. They would adjust their visualizations to better represent their data, such as selecting which part of the data being plotted.

For example, P22 filtered the weight and height of Poodles to a particular range and then made a scatter plot. They made this decision so that the distribution of the data could be easily seen: "I think if I don't filter by weight and height, the plot is going to be messy with dots all around." In another case, P26 noticed that the scatter plot they created was "really skewed to the upper right corner." They tried to add filters to the visualized data in the hope of zooming in on the visualization, but later realized that "the filter could not change the axis, it was changing the data itself."

With no way to adjust the axes in our current design, P24 brainstormed some ideas on how such a feature could look like: "I guess, maybe either make it set to the area around the points, or have it so that the user can move the graph around or set the boundaries that they want to see."

5.3 Perspectives

In this section, we describe several key *perspectives* that participants expressed during the workshops, including **PE1:** Large datasets are incomprehensible without computational support; **PE2:** Data is shaped by human decisions; **PE3:** Data can be incomplete and "messy"; **PE4:** Outliers can impact visualization and analysis; and **PE5:** Data can be used to answer a range of questions.

5.3.1 PE1: Large datasets are incomprehensible without computational support. Participants were able to understand why computational methods are necessary with large datasets. *Dataland* offers users the opportunity to think about how to approach and work with large datasets. All but one participant in our workshop had never worked with data on a scale of thousands of rows and multiple dimensions. When seeing the data table for the first time at the beginning of the workshop, it was common for participants to be surprised by the scale of the dataset. Many participants would scroll through the data table in the hope of getting to the bottom and seeing how many rows it contained. In other cases, both of P22 and P26's first action towards the Missing Animal dataset was to use Ctrl + F to find the keyword "Poodle," before realizing that the count might not work as the dataset might be too large to render: "there will be just like too much to count the actual things by hand!" [3]

Participants also reflected on the importance of slicing a large dataset to increase the efficiency of their analysis. For example, P22 noticed that their code took "forever" to categorize Poodles compared to others, and later realized that they looped through all breeds of dogs in the dataset instead of filtering out the Poodles first.

5.3.2 PE2: Data is shaped by human decisions. Participants were also able to speculate about how the data might be collected and how human decisions in the data collection process might impact and shape the data. For example, P23 was aware that the Poodle dataset was created by humans and commented that "someone is tracking the dogs and this data is probably collected by shelters." When asked about how each column of the Poodle data was generated, P27 guessed the human activities involved in the process:"[for location,] there might be a tag around the foot... For age, we can look at the teeth. Other information was gathered from observations like color, sex, etc." P25 further speculated on issues that might occur in data collection: "Data is registered by human and they might not enter accurate information. For example, they may not be certain about the breed of the dog." These perspectives also influenced how the participants approached and interpreted visualizations and analysis results.

5.3.3 PE3: Data can be incomplete and "messy". Participants were able to recognize that there could be missing values and mistakes in data and that they needed to make decisions on how to deal with it. In the datasets used in the workshops, there were some blank values or missing data. Some participants were able to notice

[3] To improve performance, *Dataland* renders only the visible part of the table at a given point in time, thus making the Ctrl-F approach not work.

this missing data when first scrolling through the data table and expressed curiosity: "what do we do about missing data?" (P12) We left it open-ended and let participants decide on their own how to deal with missing data. Some participants, like P3 , created programs to filter out the missing values; others, such as P10, kept these missing values as "nulls" (our terminology) and included them when making visualizations. P10 specifically created a bar plot that counted the number of penguins on each island, and found that the null values were plotted as zero. Initially thinking of it as a "bug" in the visualization code, P10 were able to speculate on alternative ways of dealing with null values in visualization.

5.3.4 *PE4: Outliers can impact visualization and analysis.*

Participants were also able to understand what outliers were in the data, pay attention to them, and reflect on the implications of the existence of these outliers. The storylines included tasks that guided learners to look for outliers. For example, in the Penguin A storyline, one of the tasks was to locate penguins with a "very flat" bill (i.e., outliers in the ratio of length and width). Some participants, like P14, were initially confused and asked "how do we know what is flat [bills] and what is not flat? We need a definition!" (P14) After plotting the bill length and width on a scatter plot, P14 were able to see the shape of the data and successfully locate the outliers. Participants also thought critically about outliers. For example, P24 critically questioned the data collection process after seeing some extraordinary weight in the Poodle dataset: "The outliers are weird, like they are doing it's own thing over there", and also guessed that it was caused by human errors in data entry. P26 further thought about how to best deal with outliers in analysis, making a plan to filter out outliers to better visualize the data.

5.3.5 *PE5: Data can be used to answer a range of questions.*

Our final perspective addresses perhaps what the key aim of working with data is, i.e., to answer questions and to recognize that many different questions can be asked and answered with a dataset. In the workshop activities that we designed, most questions were preset by us. However, even in such a scenario, at least some of our participants realized that the questions that can be answered go beyond what we had provided. For example, P26 noticed that some columns of the dataset were not used in the Poodle storyline and came up with their own questions that they would like to answer with the same dataset: "Is there a specific kind of doggy that are more easily to get missing? Are there a location that they are more likely to go? Are there things that makes them more likely to be missing?" More generally, P4 reflected that "with datasets, you can do so much stuff. You can categorize data in so much different ways." Participants also imagined new projects that they could do with *Dataland*. Those projects represented topics that they were interested in and relevant to their lives. Here are a few examples that the participants came up with:

> "It will probably be used mostly for like, at least in my end, organizing stuff. I have a lot of stuff I need to keep track of. So if you have a lot of Lego bricks, for example, making sure that I know how many Lego bricks I have, which kind." (P3)

> "Dataset can be used for getting information about shoe sales. I can use it to gather information about whether to buy or sell shoes" (P27)

However, we also observed some cases where young participants were constrained by *Dataland* when imagining broader possibilities of data. Since both of our storylines focused on finding lost animals, when asked to imagine what other projects can be done with *Dataland*, some young participants proposed projects very similar to what we did in the workshop: "[maybe] someone lost its way home... You need to solve how he should go home.[. . .] [researcher probed for what other data that they can imagine] Maybe like rabbits, bunny, cat, dogs. . . " (P2) This suggests that we may need to provide more open-ended, exploratory activities along with the necessary support to learners.

6 DISCUSSION

In this section, we synthesize our findings to (a) provide an initial framework for studying and developing computational data literacy, and (b) offer recommendations for designers of block-based data programming systems for young learners.

6.1 A framework for studying and developing computational data literacy

In §5, we build on [8] and describe a framework for computational data literacy that involves understanding new concepts, adopting new practices, and acquiring new perspectives through programming with data. It is crucial to clarify here that we do not claim that programming is the only pathway to these concepts, practices, and perspectives. Rather, our findings demonstrate how scaffolding learners to program with data can facilitate the acquisition of these elements. Furthermore, many aspects of our framework are not merely new additions, but ones that may require a shift from pre-existing knowledge. For example, **PR1** represents a shift from using a combination of conditionals and loops—things that operate on individual data points—to procedures that operate at the level of a collection of data points. We hypothesize that this phenomenon of the need to shift from pre-existing CT knowledge will also be evident in aspects of computational literacy that we have not explored in this paper, such as the concept of vector operations with data. Additionally, the concepts that we surfaced in §5.1 overlap with and complement the list of competencies for children to reason with data outlined in Rubin [42]. As several of this broader set of concepts are traditionally recognized as key and challenging in data science education, our findings indicate that children are able to construct their own understandings of those concepts with a system like *Dataland*. Several practices that we observed in our workshops also echo the exploratory and iterative workflow of professional data analysts [11, 27], and our findings provide an unique perspective on how children can organically build up those practices off their preexisting knowledge on programming and data. Furthermore, several perspectives that we observed from our workshops speak to the calls for promoting critical data literacy among children [32]. In particular, **PE2**, **PE3**, and **PE4** address known perceptual gaps about data that exist among children, such as whose and what decisions are involved in data analysis [49].

That said, we do not claim that these represent a comprehensive view—our findings are ultimately mediated and limited by our tool and pedagogical approach, such as limiting data structures to simple scalar variables, or not supporting remixing, which has been studied as a key practice for computational data literacy [56]. Similar limits also apply to the original framework by Brennan and Resnick [8], who were largely informed by empirical evidence from the Scratch programming language and online community. We offer these additions to the framework as a starting point for scholars and practitioners interested in computational data literacy, with the hope that new tools and pedagogical approaches will develop the framework even further.

6.2 Design implications for data programming systems for youth

6.2.1 Building on advantages of block-based editing, but for data. The design of data programming systems should support its user to easily program and tinker with data. This echos the design principle of "low floor" [41]. Our design of *Dataland* offers an example of this principle in action. While it is already known that the visual block-based language lowers the burden for learners to remember the syntax of a programming language [3], for data in particular, fixed drop-down menus with data column titles further lowered the burden of memorizing and typing names. The shapes and snapping interaction of visual blocks also provides hints on the compatibility of different data operations. Furthermore, in *Dataland*, learners can easily see the immediate result of their analysis programs to identify and fix issues within the process. All these designs were crucial for learners to construct their own understanding of unfamiliar data concepts through tinkering and experiments.

6.2.2 Choosing and using black boxes carefully and intentionally. Building on Resnick and Silverman's [41] principle of "choosing black boxes carefully,", we saw it play out in specific ways for programming with data. For example, we explicitly wanted learners to know about filtering, and hence included a `filter` block, rather than making learners implement equivalent functionality through loops and conditionals. This type of design choice also echos the design principle that learner-centered data analytic systems should offer learning-centered scaffolds on specific concepts [15]. However, we did see instances where learners considered the use of conditional blocks (§5.2.1) versus the `filter` block. This suggests that the question of black boxes applies not just to the design phase, but also to the learning or use phase, and it is possible to have a useful discussion of when it might be useful for the learner to rely on the black box, vs. when opening up the black box might make sense.

6.2.3 Guidance and creativity. Finally, the design of data programming systems for young people should provide learners with guidance on how to approach data while leaving enough space for creative explorations. Without adequate support, novices find it difficult to explore a large dataset in a meaningful way [5, 55]. Echoing prior design knowledge about supporting children to inquire with data and expand their inquiries [54], in the design of *Dataland*, we guided learners through a series of questions in storylines and at the same time, we left the specific solutions open for participants to figure out. Most participants found this arrangement to be effective as they did not have to start the analysis from scratch while still having the agency to figure out specific implementations. But some participants also commented that *Dataland* reminded them of school assignments where they simply followed the instructions and had limited space for innovation, citing that they could not develop their own questions and analysis plan with the data. This type of functional fixation is a limitation in our design, and echos other studies on children's learning of computational concepts in constructionist systems [9]. Another limitation of our paper is that our study does not fully address the broader social and cultural contexts of data and data literacy, as called for by Lee et al. [32] on humanistic approaches to data science education. We call for future designs to consider these issues and explore ways to provide learners with sufficient guidance while encouraging them to form and answer questions with data in their own contexts.

7 CONCLUSION

In this paper, we presented *Dataland*, a visual block-based programming system designed to promote youth computational data literacy. Having 27 children tried out our system to process, analyze, and visualize data in our workshops, we uncovered a series of concepts, practices, and perspectives that emerged from their engagement with data. Building on prior work, we contribute a taxonomy that describes computational data literacy along those aspects and a series of recommendations for future designers of systems to support the development of computational data literacy. We hope that our work will contribute to empowering young people to learn with and about data in powerful new ways.

ACKNOWLEDGMENTS

Some of the text and images in this paper are based on a prior short paper [33] that accompanied a technology demo. We are grateful to Chris Lyu, Mari Woodworth, and Shivam Hingorani for their assistance with this project. Financial support for this work came from the U.S. National Science Foundation (grants #1948113 and #2230291), the University of Washington, and the University of North Carolina at Chapel Hill.

8 SELECTION AND PARTICIPATION OF CHILDREN

Our research protocol was approved by the Institutional Review Board (IRB) that reviews and oversees human subjects research in our institution. Parental consent and participant assent were obtained for every participant. Assent forms were written using an age-appropriate language. Consent and assent forms were approved by our IRB prior to the study.

8.1 Information about the participants and recruitment

We recruited participants through social media posts, email lists of local educators, our own personal and professional contacts, etc., which resulted in 27 participants, all under 18 years old. In the first 4 workshops (May and June), we set the age range to be 8 to 17 years old. After noticing that at least one of the younger participants had trouble understanding some of the concepts being covered and potentially needing a different storyline tailored for their age group, in the remaining 3 (November) workshops, we further constrained the age range to 12 to 17 years old. We did not collect gender identities of the participants.

We had participants with a diverse range of experience in programming and mathematics. We had a total of 15 participants with experience in Scratch. 10 participants had experience with a text-based programming language such as Python, Java, and JavaScript. Two participants had taken the high school AP computer science course, and three participants had taken other computer programming courses. One participant did not have prior programming experience. Most participants had taken basic middle school and high school level mathematics classes such as Pre Calculus. Two participants had taken the advanced high school AP Statistics course. One participant had taken a college-level introductory data science class.

8.2 Participation in *Dataland* workshops.

All participants participated in at least one *Dataland* workshop. In all workshops, participants were explained what a research project is and were informed that all participation was voluntary. Before the start of each workshop, participants were given a randomly generated username and password pair to log into the *Dataland* system, as well as tutorial videos and handouts to familiarize participants with the programming environment. Each workshop started with an introduction to the *Dataland* system, the datasets, and the system. After the introduction, participants were instructed to interact with the *Dataland* system and follow a storyline to conduct data analysis and visualization. Participants were asked to follow the storyline and complete as many tasks in the storyline as they could at their own pace. Participants were advised to take a break every 35 minutes.

Each workshop was facilitated by members of the research team. During the workshop, facilitators would walk around the room, answering questions and offering help. For the sessions with low attendance (e.g., the workshop sessions with 1 or 2 participants), the facilitators would sit next to the participants and answer questions by request. In all workshops, the facilitators would also observe and take field notes on how the participants approached the tasks and engaged with the system, asking them about the decisions they made, their thoughts on certain parts of the analysis, and any challenges they encountered.

At the end of the workshop, participants were offered an optional interview with a facilitator, where the interview questions focused on the overall experiences of the participants with *Dataland*, their reflection on the process of working with data, and any feedback and advice they had on improving *Dataland*. Participants were also asked about their previous experiences with programming and mathematics in other contexts (e.g., at school), and how working with *Dataland* compared with those previous experiences.

REFERENCES

[1] Austin Cory Bart, Ryan Whitcomb, Dennis Kafura, Clifford A. Shaffer, and Eli Tilevich. 2017. Computing with CORGIS: Diverse, Real-world Datasets for Introductory Computing. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. ACM, Seattle Washington USA, 57–62. https://doi.org/10.1145/3017680.3017708

[2] Satabdi Basu, Betsy Disalvo, Daisy Rutstein, Yuning Xu, Jeremy Roschelle, and Nathan Holbert. 2020. The Role of Evidence Centered Design and Participatory Design in a Playful Assessment for Computational Thinking About Data. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 985–991. https://doi.org/10.1145/3328778.3366881 event-place: Portland, OR, USA.

[3] David Bau, Jeff Gray, Caitlin Kelleher, Josh Sheldon, and Franklyn Turbak. 2017. Learnable Programming: Blocks and Beyond. *Commun. ACM* 60, 6 (May 2017), 72–80. https://doi.org/10.1145/3015455 Place: New York, NY, USA Publisher: Association for Computing Machinery.

[4] Burcu Berikan and Selçuk Özdemir. 2020. Investigating "Problem-Solving With Datasets" as an Implementation of Computational Thinking: A Literature Review. *Journal of Educational Computing Research* 58, 2 (April 2020), 502–534. https://doi.org/10.1177/0735633119845694

[5] Mary Beth Kery and Brad A. Myers. 2017. Exploring exploratory programming. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 25–29. https://doi.org/10.1109/VLHCC.2017.8103446

[6] Rahul Bhargava, Ricardo Kadouaki, Emily Bhargava, Guilherme Castro, and Catherine D'Ignazio. 2016. Data murals: using the arts to build data literacy. *The Journal of Community Informatics* 12, 3 (2016).

[7] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[8] Karen Brennan and Mitchel Resnick. 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. 25.

[9] Ruijia Cheng, Sayamindu Dasgupta, and Benjamin Mako Hill. 2022. How Interest-Driven Content Creation Shapes Opportunities for Informal Learning in Scratch: A Case Study on Novices' Use of Data Structures. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 228, 16 pages. https://doi.org/10.1145/3491102.3502124

[10] Tamara Clegg, Daniel M Greene, Nate Beard, and Jasmine Brunson. 2020. Data Everyday: Data Literacy Practices in a Division I College Sports Context. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[11] Anamaria Crisan, Brittany Fiore-Gartland, and Melanie K. Tory. 2020. Passing the Data Baton : A Retrospective Analysis on Data Science Work and Workers. *IEEE Transactions on Visualization and Computer Graphics* 27 (2020), 1860–1870.

[12] Sayamindu Dasgupta and Benjamin Mako Hill. 2017. Scratch Community Blocks: Supporting Children as Data Scientists. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM Press, 3620–3631. https://doi.org/10.1145/3025453.3025847 event-place: New York, New York.

[13] Erica Deahl. 2014. Better the data you know: Developing youth data literacy in schools and informal learning environments. *Available at SSRN 2445621* (2014).

[14] Catherine D'Ignazio. 2017. Creative data literacy: Bridging the gap between the data-haves and data-have nots. *Information Design Journal* 23, 1 (2017), 6–18. Publisher: John Benjamins.

[15] Catherine D'Ignazio and Rahul Bhargava. 2016. DataBasic: Design principles, tools and activities for data literacy learners. *The Journal of Community Informatics* 12, 3 (2016).

[16] Catherine D'Ignazio and Lauren F Klein. 2020. *Data feminism*. Mit Press.

[17] Paul Dourish and Edgar Gómez Cruz. 2018. Datafication and data fiction: Narrating data and narrating with data. *Big Data & Society* 5, 2 (July 2018), 2053951718784083. https://doi.org/10.1177/2053951718784083

[18] Catherine D'Ignazio and Rahul Bhargava. 2015. Approaches to Building Big Data Literacy. In *Proceedings of the Bloomberg Data for Good Exchange Conference 2015*. https://dam-prod.media.mit.edu/x/2016/10/20/Edu_D'Ignazio_52.pdf event-place: New York, N.Y..

[19] Juliana Elisa Raffaghelli. 2020. Is Data Literacy a Catalyst of Social Justice? A Response from Nine Data Literacy Initiatives in Higher Education. *Education Sciences* 10, 9 (2020), 233. Publisher: Multidisciplinary Digital Publishing Institute.

[20] Melanie Feinberg. 2017. A Design Perspective on Data. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, Denver, Colorado, USA, 2952–2963. https://doi.org/10.1145/3025453.3025837

[21] William Finzer and Dan Damelin. 2015. Building the CODAP Community. *@Concord* 19, 2 (2015), 8–9. https://concordconsort.wpenginepowered.com/wp-content/uploads/2016/12/newsletters/2015/fall/at-concord-fall-2015.pdf

[22] N. Fraser. 2015. Ten things we've learned from Blockly. In *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*. 49–50. https://doi.org/10.1109/BLOCKS.2015.7369000

[23] Samantha Hautea, Sayamindu Dasgupta, and Benjamin Mako Hill. 2017. Youth Perspectives on Critical Data Literacies. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM Press, 919–930. https://doi.org/10.1145/3025453.3025823 event-place: New York, New York.

[24] Benjamin Mako Hill, Dharma Dailey, Richard T Guy, Ben Lewis, Mika Matsuzaki, and Jonathan T. Morgan. 2017. Democratizing Data Science: The Community Data Science Workshops and Classes. In *Big Data Factories: Scientific Collaborative Approaches for Virtual Community Data Collection, Repurposing, Recombining, and Dissemination*, Nicolas Jullien, Sorin A. Matei, and Sean P. Goggins (Eds.). Springer Nature, New York, New York, 115–135.

[25] Allison Marie Horst, Alison Presmanes Hill, and Kristen B. Gorman. 2020. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data.* https://doi.org/10.5281/zenodo.3960218

[26] Yasmin Kafai. 2017. Connected Gaming: An Inclusive Perspective for Serious Gaming. *International Journal of Serious Games* 4, 3 (Sept. 2017). https://doi.org/10.17083/ijsg.v4i3.174

[27] Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer. 2012. Enterprise Data Analysis and Visualization: An Interview Study. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (Dec. 2012), 2917–2926. https://doi.org/10.1109/TVCG.2012.219

[28] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, and others. 2016. Jupyter Notebooks-a publishing format for reproducible computational workflows.. In *ELPUB*. 87–90.

[29] Sean Kross and Philip J. Guo. 2019. Practitioners Teaching Data Science in Industry and Academia: Expectations, Workflows, and Challenges. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–14. https://doi.org/10.1145/3290605.3300493

[30] Victor Lee and Michelle Wilkerson. 2018. Data Use by Middle and Secondary Students in the Digital Age: A Status Report and Future Prospects. (2018). https://digitalcommons.usu.edu/itls_facpub/634 Commissioned Paper for the National Academies of Sciences, Engineering, and Medicine, Board on Science Education, Committee on Science Investigations and Engineering Design for Grades 6-12. Washington, D.C..

[31] Victor R. Lee. 2013. The Quantified Self (QS) Movement and Some Emerging Opportunities for the Educational Technology Field. *Educational Technology* 53, 6 (2013), 39–42. http://www.jstor.org/stable/44430216 Publisher: Educational Technology Publications, Inc..

[32] Victor R. Lee, Daniel R. Pimentel, Rahul Bhargava, and Catherine D'Ignazio. 2022. Taking data feminism to school: A synthesis and review of pre-collegiate data science education projects. *British Journal of Educational Technology* 53, 5 (Sept. 2022), 1096–1113. https://doi.org/10.1111/bjet.13251

[33] Lingyu Wang and Sayamindu Dasgupta. 2022. Dataland: An Informed, Situated, and Critical Approach to Data Literacy. In *General Proceedings of the 2nd Annual Meeting of the International Society of the Learning Sciences 2022*. International Society of the Learning Sciences, Hiroshima, Japan.

[34] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. The Scratch Programming Language and Environment. *ACM Trans. Comput. Educ.* 10, 4 (Nov. 2010). https://doi.org/10.1145/1868358.1868363 Place: New York, NY, USA Publisher: Association for Computing Machinery.

[35] Camillia Matuk, Susan Yoon, Joseph Polman, Anna Amato, Jacob Barton, Nicole Marie Bulalacao, Francesco Cafaro, Lina Chopra Haldar, Amanda Cottone, Krista Cortes, and others. 2020. Data Literacy for Social Justice. *International Society of the Learning Sciences (ISLS)* (2020).

[36] Seymour Papert. 1980. *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books, New York, NY. https://catalog.lib.unc.edu/catalog/UNCb1682803

[37] Miranda C. Parker and Leigh Ann DeLyser. 2017. Concepts and Practices: Designing and Developing A Modern K-12 CS Framework. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 453–458. https://doi.org/10.1145/3017680.3017778 event-place: Seattle, Washington, USA.

[38] Yim Register and Amy J Ko. 2020. Learning Machine Learning with Personal Data Helps Stakeholders Ground Advocacy Arguments in Model Mechanics. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*. 67–78.

[39] Mitchel Resnick. 2016. Designing for Wide Walls. https://design.blog/2016/08/25/mitchel-resnick-designing-for-wide-walls/ Archived at https://perma.cc/9N3P-48E3.

[40] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (Nov. 2009), 60–67. https://doi.org/10.1145/1592761.1592779

[41] Mitchel Resnick and Brian Silverman. 2005. Some Reflections on Designing Construction Kits for Kids. In *Proceedings of the 2005 Conference on Interaction Design and Children (IDC '05)*. ACM, New York, NY, USA, 117–122. https://doi.org/10.1145/1109540.1109556

[42] Andee Rubin. 2020. Learning to Reason with Data: How Did We Get Here and What Do We Know? *Journal of the Learning Sciences* 29, 1 (2020), 154–164. Publisher: Taylor & Francis.

[43] Andee Rubin, James Hammerman, and Cliff Konold. 2006. Exploring informal inference with interactive visualization software. In *Proceedings of the Seventh International Conference on Teaching Statistics*. International Statistical Institute Voorburg.

[44] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2017). http://idl.cs.washington.edu/papers/vega-lite

[45] Mabel F. Stryker. 1923. *Little dog Ready: how he lost himself in the big world.* H. Holt and Company, New York. OCLC: 1157168860.

[46] Rabindranath Tagore. 1930. *Sahaj Path - Part 1.* Visva-Bharati Granthanbibhag, Santiniketan, India. https://archive.org/embed/SahajPath-Part1-Bangla

[47] Alan Freihof Tygel and Rosana Kirsch. 2016. Contributions of Paulo Freire for a Critical Data Literacy: a Popular Education Approach. *The Journal of Community Informatics* 12, 3 (Oct. 2016). https://doi.org/10.15353/joci.v12i3.3279 Number: 3.

[48] Volodymyr Agafonkin. 2023. Leaflet — an Open-Source JavaScript Library for Interactive Maps. https://leafletjs.com/.

[49] Ge Wang, Jun Zhao, Max Van Kleek, and Nigel Shadbolt. 2022. 'Don't make assumptions about me!': Understanding Children's Perception of Datafication Online. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (Nov. 2022), 1–24. https://doi.org/10.1145/3555144

[50] David Weintrop, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. 2016. Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology* 25, 1 (Feb. 2016), 127–147. https://doi.org/10.1007/s10956-015-9581-5

[51] Michelle Wilkerson, William Finzer, Tim Erickson, and Damaris Hernandez. 2021. Reflective Data Storytelling for Youth: The CODAP Story Builder. In *Interaction Design and Children (IDC '21)*. Association for Computing Machinery, Athens, Greece, 503–507. https://doi.org/10.1145/3459990.3465177

[52] Leland Wilkinson. 2005. *The Grammar of Graphics*. Springer New York, New York, NY.

[53] Jeannette M. Wing. 2006. Computational Thinking. *Commun. ACM* 49, 3 (March 2006), 33–35. https://doi.org/10.1145/1118178.1118215

[54] Annika Wolff, Michel Wermelinger, and Marian Petre. 2019. Exploring design principles for data literacy activities to support children's inquiries from complex data. *International Journal of Human-Computer Studies* 129 (Sept. 2019), 41–54. https://doi.org/10.1016/j.ijhcs.2019.03.006

[55] Kanit Wongsuphasawat, Yang Liu, and Jeffrey Heer. 2019. Goals, Process, and Challenges of Exploratory Data Analysis: An Interview Study. (2019). https://doi.org/10.48550/ARXIV.1911.00568 Publisher: arXiv Version Number: 1.

[56] Rabia Yalcinkaya, Hamid Sanei, Changzhao Wang, Li Zhu, Jennifer Kahn, and Shiyan Jiang. 2022. Remixing as a Key Practice for Coding and Data Storytelling. In *Proceedings of the 15th International Conference on Computer-Supported Collaborative Learning - CSCL 2022*. International Society of the Learning Sciences, Hiroshima, Japan, 407–410.