The Tools Being Used to Introduce Youth to Data Science

Peter F Moon**
pmoon@umd.edu
University of Maryland
College Park, MD, USA

Rotem Israel-Fishelson
rotemisf@umd.edu
University of Maryland
College Park, MD, USA

Rachel Tabak
retabak@umd.edu
University of Maryland
College Park, MD, USA

David Weintrop
weintrop@umd.edu
University of Maryland
College Park, MD, USA

ABSTRACT

Data is increasingly shaping the way people interact with each other and the world more broadly. For youth growing up in an increasingly data-driven society, it is critical they have foundational data literacy skills. A central component of data literacy is the ability to collect, analyze, visualize, and make meaning from data. All of these activities are mediated and shaped by the tools that youth use to carry out these data practices. Given the essential role tools play in enabling and supporting youth in engaging with and interpreting data, understanding what tools are used and how they are used in educational contexts will help us understand how youth are being prepared to be data-literate citizens. In this paper, we present the analysis of the data collection and analysis tools used in 4 widely adopted high school data science curricula. The analysis attends to both what tools are used as well as what datasets they are used to analyze. This work contributes to our understanding of the way youth are being introduced to concepts and practices from the field of data science and the role the tools play in shaping those experiences.

CCS CONCEPTS

• **Human-centered computing** → **Interaction design**; • **Applied computing** → **Interactive learning environments**.

KEYWORDS

data science education, data analysis tools, youth

ACM Reference Format:
Peter F Moon, Rotem Israel-Fishelson, Rachel Tabak, and David Weintrop. 2023. The Tools Being Used to Introduce Youth to Data Science. In *Interaction Design and Children (IDC '23), June 19–23, 2023, Chicago, IL, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3585088.3589363

*Corresponding Author

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

ACM ISBN 979-8-4007-0131-3/23/06.
https://doi.org/10.1145/3585088.3589363

1 INTRODUCTION

Youth today are growing up in a constantly connected digital world driven by technological advances and data growth. Youth constantly create and consume data as they engage in social media, complete school assignments, play video games with friends, and stay connected with wearables and smartphones. These computationally-mediated activities invite youth to express their creativity, develop social interactions, and participate in civil discourse [6]. In light of the increasing impact of data on their lives and their crucial role in society, developing data literacy is vital from an early age [22, 28]. Data literacy can help youth interpret the information they consume, make evidence-based decisions, and become more socially and civically engaged [2]. Moreover, the impact of data on youth is not necessarily uniform as people from populations historically excluded from computing (e.g., BIPOC, economically disadvantaged, women, English language learners, neurodiverse) are disproportionately adversely affected by biases and predatory uses of data-driven algorithms [3]. Thus, it is increasingly important to instill data literacy as part of K-12 education to promote equity and create data-driven citizens [39].

Data science is an evolving discipline with enormous possibilities for discovery and learning [9]. Various new data analysis tools and data science curricula have emerged in the last five years to support youth learning about and engaging with data [21]. New data analysis tools and technologies support youth in engaging in authentic data science practices that historically have only been accessible to experts [32]. The emergence of simple user interfaces and clever designs to automate complex tasks allows novices to engage in sophisticated data science practices, including data analysis and visualization [11]. The introduction of such tools gives youth opportunities for independent exploration and discovery and enables deeper interaction with the data. Further, these tools have been incorporated into new curricula designed to introduce youth to data science and teach them foundational concepts and computational practices for collecting, sorting, extracting, and analyzing data from different sources. However, since most youth have little or no prior experience with formal data science practices, their experience with the tools must be meaningful and engaging. To this end, it is important to integrate heterogeneous datasets into the curricula, both in terms of the topics they deal with, the size, and the type of data, and provide youth with opportunities to explore

and synthesize them [12]. Situating instruction in the lived experiences of youth by selecting engaging and relevant datasets can foster their agency and ownership and increase their motivation.

As a core practice of data science is manipulating and interrogating datasets, it is crucial to consider the tools youth are using to be introduced to data science, as the tools themselves play an essential role in shaping youths' experiences and emerging practices [31]. Moreover, considering the tools alongside the datasets youth are using is critical given the importance of drawing on the lived experiences of youth for creating engaging and equitable data science learning experiences [37]. However, to date, most research on data science tools for youth has focused on the design of the tools in isolation and has not considered the datasets used in conjunction with the tools [31].

In this work, we investigate the tools and the accompanying datasets used to introduce youth to the powerful ideas of data science. In particular, we analyze four of the most widespread high school data science curricula: Bootstrap:Data Science [5], CodeHS [8]. Introduction to Data Science [17], and YouCubed Explorations in Data Science [42]. In performing this analysis, we broaden the focus of research on youth and data science from attending to data analysis tools in isolation to more broadly considering what youth do with those tools and the dataset they are using. More specifically, we pursue the following research questions:

RQ1. What tools are being used to introduce youth to data science?

RQ2. What types of datasets are youth using when engaging with data science tools?

2 LITERATURE REVIEW

2.1 Youth and Data Science

Youth create and consume vast amounts of information. Their online activities leave digital traces when they watch videos, send messages or photos to friends, pay with an app, navigate to a site, and even go to a doctor's appointment. Youth have different interpretations of the nature of the data, but they often do not fully understand how it affects their personal lives directly [4, 19]. Thus, data literacy is necessary for youth to better interpret the information around them and make evidence-based decisions in a data-driven world [22].

This growing demand for data literacy, driven by the increasing influence of data science in academia, industry, and society, is leading to its growing presence in primary and secondary education [17]. Dedicated curricula have been developed in recent years by leading universities and organizations with the aim of introducing youth to the field of data science [10, 28].

Data science curricula use various analytical and visualization technologies and tools to enable the exploration of phenomena and engagement with questions arising from datasets in fields ranging from health and politics to entertainment and sports [35]. Such tools allow the application of computational techniques for collecting, storing, retrieving, and analyzing data [17]. The data analyzed by these tools can come from various sources, including data collected from public databases, data collected in automatic forms using sensors and input devices, data generated algorithmically from online environments, and fictitious data invented for pedagogical needs. The data are often ripe for analysis, and youth do not have the opportunity to participate in their curation [11, 26].

Data science curricula have an important role in promoting civic responsibility and developing critical thinking about how data are collected, produced, and used. Curricula can help youth understand their role as producers and consumers of data and the dangers of using data without considering its implications on individuals, organizations, and society [3, 27, 35]. Also, they must teach how to use analytical tools responsibly and ethically [15]. Therefore, it is vital to examine the state of the curricula and study which tools are available and which data manipulations youths are required to perform with them.

2.2 Data Science Tools

Introducing technologies into data science curricula expands the opportunities for youths to collect and explore data independently inside and outside the school walls [11]. Data collection and analysis tools can be divided into four broad categories: spreadsheets; visual interfaces; scripting languages; and other interfaces [31]. Spreadsheets are commonly used in K-12 instruction because they are included in software packages (e.g., Google Sheets) and, therefore, freely available for use. Spreadsheets help in documenting data that youths can collect on their own. Also, they support performing simple manipulations on datasets, such as filtering by values, deleting records, and displaying basic statistics in their raw form and in a visual display of charts and graphs [1, 27].

Visual tools provide graphical user interfaces that include menus or drag-and-drop features and are often designed for educational contexts (e.g., CODAP, Tuva). Tinkerplots [38], Fathom [14], SimCalc [33], and NetLogo [41] are among these tools, which often include sample datasets for students to explore. These tools support functions for organizing quantitative and qualitative data in tables, graphs, and other visual representations to explore patterns in a dataset without any programming skills [11]. The friendly interface offers interactivity which supports the transformation of data representations and exploratory analysis of the more complex data from different angles [31].

Scripting languages, such as R, Python, and Pyret, are used by data scientists for data analysis; they are, therefore, the most functional - but often have a steep learning curve. Despite their complexity compared to other analysis tools, these languages are adopted in educational contexts because they allow the automation of advanced functions on large datasets [22, 31].

Other interfaces include common commercial tools (such as SPSS) and environments like YouCubed [42] and Google Colab, which provide scaffolded interactive frameworks for performing step-by-step computational operations by modifying given codes or writing codes incrementally [23, 31].

The analysis and visualization tools currently used in data science introductory curricula are diverse and vary in their capabilities. A recent survey examining the current state of data science in 69 colleges and universities found no uniformity regarding the technologies used in data science courses. However, it highlighted that a handful of tools are more common than others, including RStudio, Jupyter, and Excel. Preference for one tool over another stems from pedagogical considerations, the tool's functionality, and the

technological barriers of the teaching personnel [36]. There are a variety of tools offered to introduce data science to K-12. However, discussing the tools themselves is not enough since examining the context in which they are embedded is just as important [31]. In this study, we examine the tools and datasets used in well-established curricula and analyze how youths engage and interact with them during their studies.

3 METHODS

In this section, we present our approach to answering the above research questions. First, we explain how we selected the four data science curricula at the center of our study, and then briefly characterize these curricula. Next, we present our analytic approach, outlining the dimensions we used to understand (1) the varying technological tools utilized in these curricula, and (2) the datasets that students explore in these curricula.

3.1 Focal High School Data Science Curricula

To identify the focal curricula for our study, we first reviewed the curricular resources on the DataScience4Everyone website. DataScience4Everyone - a coalition of policymakers, industry leaders, schools, and scholars - outlines 12 curricula for high school: Bootstrap:Data Science (BS:DS), Code.org, CodeHS, CourseKata, Data8, DataCamp, Education Development Center, Key2Stats, STEMcoding, Stats Medic, Introduction to Data Science (IDS), and YouCubed. In deciding which of these curricula to include, we defined four criteria for curricula: (1) it must focus primarily on data science (rather than another content area with data science interwoven); (2) it must be high-school focused; (3) it must be an actual curriculum (i.e., not a collection of activities/lessons to be curated by an educator); (4) it must be school/classroom-ready (meaning that student assignments, lesson plans, and more are provided). Multiple reviewers examined each of the 12 curricula above and then discussed each until they agreed on whether to include the curriculum in the study. Applying the four exclusion criteria narrowed the list to four curricula: Bootstrap:Data Science, CodeHS, Introduction to Data Science, and YouCubed.

Bootstrap:Data Science (BS:DS) is designed to be implemented as a standalone course or integrated into existing courses across disciplines for students in grades 7-10. Students learn the Pyret programming language in order to conduct their data analyses. The BS:DS curriculum includes 29 lessons with accompanying teacher presentation slides and student workbook pages [5]. The second curriculum, CodeHS, is a semester-long data science curriculum. It includes 58 lessons consisting of video tutorials, sample programs, programming exercises in Python, and offline handouts. CodeHS introduces students to data collection, cleaning, transformation, analysis, and visualization skills [8]. Third, Introduction to Data Science (IDS) is a year-long curriculum developed by researchers from the University of California-Los Angeles, in partnership with the Los Angeles Unified School District. It emphasizes practical data analysis to help students develop computational and statistical thinking. The fifth version, which we examined, includes four units containing 81 lessons, lab activities, practicums, and summative projects [17]. Finally, YouCubed Explorations in Data Science (YouCubed) is a project-based curriculum developed at Stanford's Graduate School of Education. Its eight units integrate a variety of tools, including Google Sheets, Python, Data Commons, and Tableau. YouCubed provides detailed lesson plans along with resources for teachers, students, and parents [42].

3.2 Analytic Approach

The four focal curricula were systematically analyzed by a team of researchers to identify every tool and dataset that youth would encounter. Each unit and activity was qualitatively analyzed, attending to what data was present and how learners engaged with it (i.e., what tool was used). After identifying each tool and dataset, two research team members independently conducted their analyses of both. For the tools identified, researchers categorized what the tool was used for (e.g. data collection, data analysis), the specific technology being used (e.g., CODAP, RStudio), and in the case where programming was involved, what language was used. For the datasets, researchers evaluated the datasets' size, proximity, and recency. After completing the analysis, the researchers compared results and measured inter-rater reliability using Cohen's Kappa (Cohen, 1960), which yielded a satisfactory coefficient of 0.8. All discrepancies were discussed and resolved as a group. Below, we discuss each analytic dimension in greater detail.

3.3 Tools Coding Scheme

Although students did not need to use any sort of technological tools to analyze data visualizations, analysis of both raw data and student-generated data required that students employ a variety of tools, environments, and languages. In determining our categories, our primary consideration was the manner of youth engagement with the tools, rather than focusing on specific interface features.

We encountered three broad categories of tools: data-gathering tools, programming, and visual analysis tools. The first category, data-gathering tools, enables youth to collect their own data for analysis rather than relying on provided datasets. In IDS, for instance, students develop a research question about water usage, and then investigate that question by observing patterns of water usage in their neighborhoods over the course of a month, inputting data into an online participatory sensing campaign manager. Students also gather data for analysis via databases. For instance, students in YouCubed, students develop a ranked list of places they might like to live using information gathered from Data Commons, which aggregates data from a wide range of sources.

To engage with a dataset (whether it is collected or provided), youth often rely on programming, in the form of three different languages - R, Python, and Pyret. IDS utilizes R, a programming language for statistical computing and graphics; CodeHS and YouCubed lean heavily on Python, a high-level, general-purpose and widely used programming language; and Bootstrap relies on Pyret, a language with a Python-like syntax, which was designed for introductory programming education.

The environment in which students program varies as well. In YouCubed, students typically work in Colab, a Google app that uses Python. Colab does not require configuration and allows for easy sharing between students and teachers via "Colab notebooks," which live in Google Drive. In IDS, students complete once-a-week

labs in RStudio, an interface for coding using R. The version of RStudio that students use in IDS is available at https://tools.idsucla.org/, which also exists as a Mobile App. Programming in Bootstrap takes place in the Pyret environment, which runs in a web browser and connects to Google Drive.

Visual analysis tools, including CODAP, EduBlocks, and Tableau, support visual analysis of data, rather than programming. In CODAP, a web-based data-analysis environment designed for students in grades 5 through 14, students interact with a user interface that allows them to arrange and rearrange data by selecting menu options and dragging attribute names [7]. The web-based EduBlocks, which was also designed for educational purposes, correlates draggable "blocks" with lines of code, allowing students to explore languages such as Python and HTML [13]. YouCubed characterizes Tableau, a business analytics tool, as "a professional version of CODAP," elaborating that industry data scientists use it to make and share visualizations with others. In order to use the online version of Tableau for free, teachers must gain access to a student bulk license, thereby acquiring a list of activation keys to distribute to students.

3.4 Dataset Coding Scheme

Dataset recency, proximity, and size are at the center of the study. We characterize these dimensions below.

3.4.1 Recency. This category captures the time period the data represent - sometimes a single year (e.g., top 100 songs of 2022) and sometimes a time span (e.g., top 100 songs of the 2010's). This category is meant to describe when the data is from, not necessarily when it was collected. A dataset generated in 2020 on crop yields in the 1800s would be characterized as "Over 10 years old," rather than "Recent." When datasets cover a timespan, we use the most recent date in coding; in other words, the 1800-1899 dataset would use 1899 to determine its recency level.

3.4.2 Proximity. Proximity, which captures how the dataset relates to the learners, is a measure derived from Lee and Delaney's [25] work. Lee and Delaney proposed a 5-point scale, ranging from 0-4. Zero describes content-agnostic data and 4 captures data that students collect about themselves and their peers. Levels 0 and 1 capture fictional datasets, while levels 2-4 capture real-world data. Level 2 data is about a topic that may be familiar to some but not all students (e.g., niche topics or topics from adult contexts). Level 3 data is on a topic one could reasonably expect learners to be familiar with - but not about the learner; alternatively, level 3 data is learner-generated but not about the learners themselves (e.g. skin tones represented in a magazine). Level 4 data is learner-created or learner-generated, and is about the learners themselves. More proximal data represents an opportunity for culturally-relevant pedagogy, as higher levels of proximity – particularly Level 4, which comes from the learners themselves – reflect data that is more relevant to the learners' lives and issues important to them and their communities.

3.4.3 Size. This category depicts how many observations or entries were in each dataset (i.e., the sample size or the number of rows). We classified the datasets into five sizes: very small (<25), small (25-100), medium (101-1,000), large (1,001-10,000), and very large (>10,000).

Table 1: Tool types by curriculum and dataset.

Tool Type	BS:DS	CodeHS	IDS	YouCubed	Total
Gathering (Database)	1	5	7	3	16
Gathering (Survey)	1	0	2	4	7
Programming	40	51	27	8	126
Visual Analysis Tool	1	0	0	6	7

4 FINDINGS

The focus of this work is to better understand how youth are being introduced to data science. More concretely, we are interested in what tools are being used and how they are being used, particularly as it relates to the datasets youth are creating and exploring. To answer the first research question, we looked at the kinds of tools youth interact with across the most widely used data science curricula and analyzed what kinds of interactions they have with these tools. To address the second research question, we sought to characterize aspects of the datasets youth analyze with these tools and how different tools might see differential use across these aspects.

4.1 The Tools Youth Use When Being Introduced to Data Science

Our analysis found a mix of data gathering, data analysis, and programming tools across the four curricula. Most of the data analysis being done by youth across the four curricula is done using a programming language (Table 1). Additionally, all 4 curricula have youth use data gathering tools, including APIs, survey tools such as Google Forms, and direct access databases. We also found that despite the growing array of visual data analysis tools, only YouCubed and BS:DS had youth conduct data science inquiry using them.

The use of data gathering tools can be seen across the curricula and the presence of learner-generated datasets. This was sometimes done by using tools to collect data directly from classmates and/or community members. For example, in YouCubed, students used a Google Form to collect a list of their peers' favorite songs. Other times, students used tools to gather data from pre-existing databases. For example, YouCubed has a lesson where students use Python to draw data from the Data Commons public Application Programming Interface (API) into a Google Sheet, while IDS has students collect data from social media websites directly through a web browser.

Across the four curricula, a relatively small number of activities had students using non-programming visual analysis tools. YouCubed contained most of these activities, several of which used CODAP, a browser-based tool where students can generate visualizations and data summaries by clicking and dragging [7]. YouCubed had activities where youth use Tableau, a business intelligence analytics tool, to help students generate visuals on water usage (with data drawn from the EPA, weather, and census data).

4.2 How Youth Program in Introductory Data Science Contexts

Programming was by far the most common tool used for the analysis of different datasets. Our analysis reveals that youth are being introduced to data science with several different environments and with different languages. Looking across the four curricula, we find that each curriculum uses only one programming language throughout. IDS uses a web-based version of RStudio, in which students write and run code written in R to analyze 27 datasets. BS:DS uses the Pyret programming language and has students write code in a web-based Pyret editor designed with support to help novice programmers and includes 41 distinct datasets. CodeHS uses its own integrated development environment (IDE) to introduce youth to data analysis with Python using a total of 51 datasets across the curriculum. Finally, YouCubed has students program in two environments, Colab (7 datasets) and EduBlocks (3 datasets), with Python as the underlying programming language for both. EduBlocks is noteworthy as it is a graphical block-based programming environment, which is an increasingly common way to introduce students to programming [40].

We find a spectrum in terms of the types of assignments and activities youth engage with across these programming languages and curricula. Sometimes, programming activities involve students running provided programs with minimal or no modifications being made by the youth. For example, in YouCubed, students run provided pieces of code in the Colab environment to analyze Small datasets about their classmates' favorite songs. In this case, the concept being explored (training and testing a prediction model) is relatively complex and writing the Python code (somewhere in the range of 100 lines) would likely be beyond the abilities of an average high schooler, especially if they had little prior programming experience before enrolling in a data science course. Pre-written programs can allow students to have hands-on experience with complex data science tasks while observing how that task would be carried out in a programming environment.

Other programming activities we analyzed gave students a starter program and asked them to either complete it or modify or customize what was provided. This approach is consistent with the Use -> Modify -> Create pedagogical sequence common to introductory programming contexts [16, 24]. IDS, for example, commonly scaffolds programming activities by providing partially-complete programs with blanks to be completed by the youth as part of the activity. For example, IDS' RStudio lab activity has learners investigate a Medium dataset of Horror movie characters and their in-movie outcomes. In this activity, students must put variable names in the correct order to complete partially written lines of code to correctly calculate statistics about the dataset. This strategy provides a halfway point where the youth are engaged in authentic data science practices using programs but do not have to do the coding completely independently. BS:DS provides a version of this type of scaffolding but does so by providing a text-based description of ways to analyze a given dataset with Pyret and providing short snippets of code for youth to then incorporate into their analysis. The idea with this approach is to minimize the need to memorize specific commands or syntax and instead focus on conceptually scaffolding data exploration.

Table 2: The ACS and ATU datasets.

Dataset	Curric.	Size	Prox.	Recency	Tool
ACS	YouCubed	V. Large	2	Recent	CODAP
ATU	IDS	V. Large	2	Not Relevant	RStudio

A third type of programming activity asks students to author programs independently, often using block-based programming tools to help support the youth. In YouCubed, users are asked to write a song shuffler program in EduBlocks to generate a sample of songs from the class favorites list. While students receive some starter blocks in their program, this programming activity is largely independent and flexible with respect to output (which does not need to be in a particular format).

4.3 How Students Explore Data with Programming and Visual Analysis Tools

Programming and analysis tools often offer different experiences when interacting with data. Because programs and functions need to be written before they run, when you conduct analyses via programming, the activities are often highly scaffolded and often prescriptive. One consequence of overly scaffolded or scripted analysis is that all students produce the same, or at least closely related, outcomes from their analysis, be it a data visualization or particular insight. Because visual analysis tools usually do not require as much technical setup or scaffolding to conduct exploratory analysis, they provide an opportunity to perform a more open exploration of a dataset.

We illustrate this difference through two activities involving datasets from the curricula we analyzed: one on data collected from the American Community Survey (ACS) from YouCubed and the other on the American Time Use (ATU) survey from IDS. The characteristics of these datasets are presented below (Table 2).

These datasets are similar in size and relevance to youth. Both consist of data from a survey administered to a very large sample of American citizens. However, the tools used for exploratory analysis are different, which in turn, results in different types of engagement with data science practices.

In the analysis of the ATU dataset, youth first view the data using pre-written commands. They then run several more commands to clean the data: some of this code is pre-written, with a few fill-in-the-blank or "write similar code using the example" prompts and a final section that asks students to write and run some new commands independently (Figure 1). In moving through this activity, youth complete a well-scaffolded lesson, running pre-existing commands, then completing partially complete commands, and finally writing new commands, and in doing so, are introduced to some important aspects of how R treats variables and data. However, they do not have any agency in terms of what questions to pursue and analyses to conduct; rather, they are carrying out assigned tasks, mostly through provided lines of code.

In contrast, the ACS activity uses CODAP, a visual data analysis tool, and results in allowing for youth-driven exploration. Rather than specifying particular tasks to run, the prompt is simply "Explore the data by looking at topics and making visuals of the data.

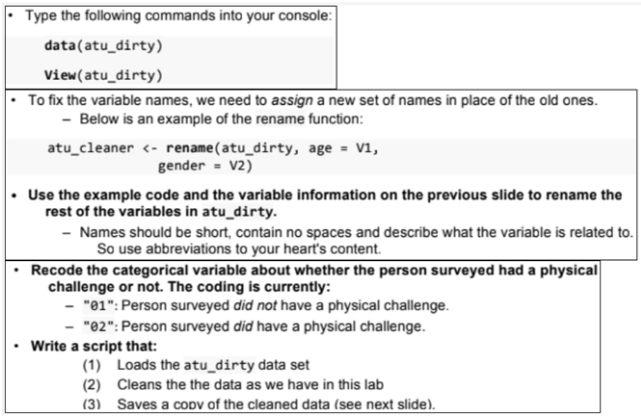

Figure 1: Examples of code task prompts in the IDS ATU data exploration.

What is interesting about the data? What would you want to learn more about? What questions do you have? Which variables are especially interesting?" [42]. This difference in approach is made possible by the features of each tool and how analysis is supported. CODAP's drag-and-drop graph creation and menu-based graphical interface (Figure 2) lower the barrier of entry for youth to conduct independent data analysis. In contrast, RStudio was designed by and for experienced data analysts and statisticians. To support youth in using RStudio, more significant scaffolds are required, thus sacrificing novice's agency during the data analysis activity.

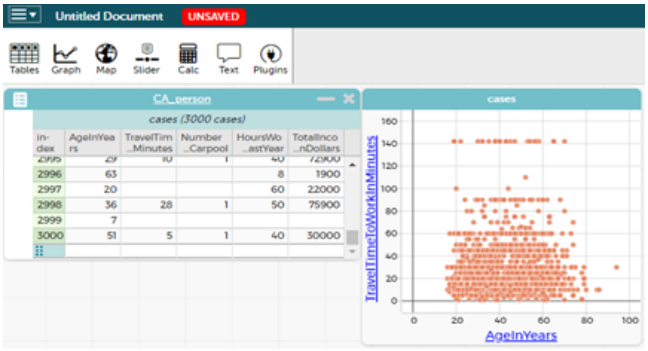

Figure 2: A CODAP scatter plot created by dragging one variable onto each axis.

4.4 What Types of Datasets do Youth Use when Engaging with Data Science Tools?

Having reviewed the different types of tools used to introduce youth to data science in high school classrooms, we now shift focus to our second research question, which investigates the datasets youth engaged with when using these data science tools. In doing so, we examine the ways that data tool selection may influence the types of datasets that can be used to introduce youth to data science. This is a consequential decision given the importance of selecting relevant and engaging datasets when introducing youth to data science,

Table 3: Frequency of dataset proximity by type of tool.

Proximity	Gathering (Database)	Gathering (Survey)	Program.	Visual Analy. Tool	Total
4	1	5	11	0	17
3	12	2	52	2	68
2	2	0	27	4	33
1	1	0	25	1	27
0	0	0	11	0	11
Totals	16	7	126	7	156

especially for youth from populations historically excluded from computing and data-intensive fields [2]. To answer this question, we look at three key characteristics of datasets: proximity, recency, and size.

4.4.1 Proximity. Looking across the 4 curricula and the 156 datasets analyzed, we find that most of the datasets analyzed by students use real data (Levels 4, 3, and 2: 118/156), with most datasets (68/156) being evaluated as a level 3 proximity (real data on a topic youth can reasonably be expected to be familiar with). However, in looking at proximity based on the type of tool being used, we do not see a correlation between the two. That is to say, the type of tool does not necessarily restrict the topics of datasets or their proximities to youth. Across the curricula (Table 3), analysis occurs mostly in programming environments regardless of proximity level, with a smaller proportion occurring in analytic tools. No visual analysis tools are used to analyze Level 4 datasets that are about the learners themselves. While most datasets at Level 4 are analyzed using programming, a few datasets at Level 4 are collected, but not used in a particular analysis.

4.4.2 Recency. Most of the datasets students analyze in these curricula comes from the past decade (85/156), and a majority of the remaining datasets are not time-relevant (either because they are comprised of fictional data, like test scores in a made-up 3rd-grade math class, or because their data does not change with time, like a dataset representing the lifespan of mammals).

Tool use is relatively consistent across different levels of recency (Table 4). The only exception is datasets collected via survey and live database data, which are by design all Fresh. A fair number of these datasets (those listed under the "Gathering" columns below) are simply collected and not analyzed by students through any means as part of the curricula. In particular, students use programming tools for about half as many analyses with Fresh data than with less recent data. At present, this feels like a missed opportunity; more programmed analysis of freshly-collected data would better represent the work done by data scientists and increase the authenticity of a data science curriculum.

4.4.3 Size. Most of the datasets analyzed (i.e. students did some part of the analysis, and were not simply presented with a premade chart or graphic summarizing the data) had less than 1,000 data entries (Medium or smaller in our coding scheme). However, when we looked at dataset size across types of tools used for analysis (Table 5), we found a clear trend: programming tools decreased in

Table 4: Frequency of dataset recency by type of tool.

Recency	Gathering (Database)	Gathering (Survey)	Program.	Visual Analy. Tool	Total
Fresh	14	7	13	1	35
Recent	0	0	25	3	28
Past Decade	0	0	21	1	22
Over 10 Years	0	0	15	0	15
Not Relevant	2	0	52	2	56
Totals	16	7	126	7	141

use as datasets got larger, while analysis tools (such as Tableau, Google Sheets, and CODAP) were used at a low but consistent rate regardless of dataset size. Overall, the major trend found was that youth are simply not working with Large (8.3%) or Very Large (7.7%) datasets very often. These findings reveal a tension with respect to providing youth with authentic data science experiences. Our analysis shows that youth often use authentic tools (i.e. programming languages and environments) but rarely have the chance to apply these tools to larger datasets.

Table 5: Frequency of dataset size by type of tool.

Size	Gathering (Database)	Gathering (Survey)	Program.	Visual Analy. Tool	Total
V. Large	7	0	3	2	12
Large	0	0	12	1	13
Medium	0	1	32	1	34
Small	2	0	27	2	31
V. Small	3	2	45	1	51
NA	4	4	7	0	15
Totals	16	7	126	7	141

5 DISCUSSION

In this work, we sought to examine the tools used to introduce youth to data science and the datasets youth engage with as part of this introduction. We investigated four well-established and popular curricula to deepen our understanding of the current state of the field and identify tensions and opportunities to improve the state of data science education. Our analysis sheds light on the various strategies for learning data science while drawing the similarities and differences in the applications of the tools and datasets. Our analysis reveals a preference for programming tools over data-gathering or visual analysis tools across the four curricula. Moreover, we identified different uses of the Use->Modify->Create pedagogical sequence in the programming tasks. Some tasks only dealt with one element of the sequence (for example, running a given code or alternatively writing an unstructured program independently), and some expressed the entire sequence. The various activities reveal opportunities to leverage the learning experience, cultivate students' independence, and achieve pedagogical goals. For example, while using pre-generated code allows a low level of autonomy, it invites students to experience performing complex manipulations on authentic datasets that they would otherwise not necessarily be able to perform on their own. This experience can strengthen students' broad understanding of the data science workflow. Experiencing activities that include youth independently programming invites a more authentic learning experience, although it depends on the skills acquired by the students and their in-depth understanding of algorithms. Curricula tend to focus on the broad aspects of data science. However, they can combine programming tasks at different independence levels to support a broad understanding of the data workflow and the algorithms, depending on the target audience and the expected skills [29].

The research results elucidate the differences between the various tools and the interactions they enable with the datasets. The programming tools prioritize text-based commands and offer broad functionality and deeper exploratory analysis. These come at the expense of interactivity and require a steep learning curve. In order to overcome these barriers and still nurture students' programming abilities, educators can prioritize tools like RStudio that provide a graphical user interface that slightly increases the interactivity and allows programming languages such as Python and R to be applied more easily [31]. In contrast, data analysis tools like CODAP allow students to perform exploratory data analysis using a friendly graphical interface that offers rich interactivity. As demonstrated, these tools often allow a quick transition between a tabular and graphical view of the data using a drag-and-drop mechanism. They also allow data aggregation but are limited in the supported statistical operations. Educators who want to develop computational and statistical thinking in students can design learning activities with tools like CODAP while implementing the different stages of the data cycle, including evaluating the existing data, their analysis, and their interpretation [17].

5.1 Tensions and Opportunities

One potential tension associated with YouCubed's approach of using many different tools is the challenge associated with introducing each new tool and the concern that students will not become adequately familiar with any one tool. On the other hand, CodeHS and IDS not including non-programming data analysis tools may be a missed opportunity as it does not capture the full breadth of the ways people encounter/use data (e.g., spreadsheets, information visualization tools). However, the focus on programming also provides a context for introducing learners to foundational computer science concepts that underpin much of data science. This tension of breadth vs. depth in terms of tool use is an open question worthy of future investigation.

When exploring the size of the datasets manipulated using the different types of tools, we see that large datasets are less common in the various curricula. While smaller datasets may be less intimidating and easier to curate for students, it is less representative of day-to-day data analysis tasks [43]. Programming tools have an inherent strength in dealing with large datasets, and curricula should play to that strength.

Examining the intersection between data recency and the types of tools used presents an opportunity to strengthen the affiliation of students to the analyzed datasets. Results show that most "Fresh" datasets were gathered by students but not further analyzed. Allowing the students to complete the full data cycle would make the learning process more proximate to the students' lived experiences. Additionally, using API (Application Programming Interfaces) for gathering data can be beneficial for situating the learning activities closer to students' interests. While API was used only once to gather data throughout the four curricula, many software and websites (e.g., Spotify, Google Maps, Twitter, YouTube, and others) offer public APIs that fit into common topics in high school data science courses and may be engaging for students.

The four curricula reviewed here reflect the state of data science education, which is constantly evolving. The study spotlights the tools integrated into the curricula and the datasets used to introduce the field to youth. Moreover, our research emphasizes the importance of the user interface, functionality, and independence offered by these tools alongside their inherent opportunities and potential.

5.2 Design Implications

Those designing curricula to introduce youth to data science should consider ways to better support live, authentic data collection. For example, we have mentioned public APIs as a possible way to include real-world data that is Fresh; another strategy could be capitalizing more on learner-generated data, such as those captured by surveys and other methods, to engage youth's interest in topics personally relevant to their interests and lives. In both cases, designers should consider ways to include larger datasets (i.e., Large or Very Large) that are freshly collected and about something the learners care about (i.e., Proximity 3-4). This approach can also support culturally-relevant pedagogy as it frames data science as a way for youth to explore issues personally relevant to them and their communities.

Designers should also consider ways to involve visual data analysis tools, which were generally underutilized throughout the four curricula we examined. Most curricula focused on a single programming language and environment, allowing youths to build familiarity with that tool throughout the course. While this focus avoids the risk of overwhelming students with many different tools to learn, designers might consider incorporating more visual data analysis tools, such as CODAP or Excel (or Google Sheets as a free alternative). If these tools are used early in the course, they could also help bridge the gap to programming, as both have spaces to write formulas and use functions to perform calculations.

Tasks should consider a low floor/high ceiling design [30]. This approach should consider what support is needed for novices to data analysis while providing room for more expert users to challenge themselves. This could include example programs or other scaffolding for programming so that newer users had something to start with, while keeping standards for the task high so that users at all levels had a challenge. It could also include visual data analysis tools or hybrid block-based/text-based programming environments, where youth could be offered the choice to perform analysis using either menus or blocks, or writing their own program in text code, depending on their comfort level. Finally, it could involve scaffolding examples of different kinds of analysis, giving youths greater independence with later examples.

Designers' final consideration should be selecting or creating an environment with a variety of built-in datasets. Environments such as CODAP and RStudio have pre-loaded datasets available that users can analyze without needing to load a dataset from a file, which can simplify the analysis process as users are learning. With various datasets available in terms of size, proximity, and recency, youths can easily gain flexible experience working with different kinds of data. This may encourage engagement from a wide range of students while providing an authentic experience representing the range of datasets a data scientist might encounter.

6 CONCLUSION

Data science is an increasingly important skill for youth growing up in the digital age, both in terms of career prospects and understanding the world around them. The tools used in data science curricula are an essential part of welcoming young users to the field, particularly in terms of what kinds of interactions they allow users to have with data. Considering what kinds of tools are used in data science curricula is thus an important consideration for designers and others who work in data science education.

7 SELECTION AND PARTICIPATION OF CHILDREN

No children participated in this work.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Grant #2141655. Any opinions, conclusions, and/or recommendations are those of the investigators and do not necessarily reflect the views of the National Science Foundation.

8 REFERENCES

[1] Bargagliotti, A., Franklin, C., Arnold, P., Gould, R., Johnson, S., Perez, L., & Spangler, D. A. (2020). Pre-K-12 guidelines for assessment and instruction in statistics education II (GAISE II). American Statistical Association.

[2] Benjamin, R. (2019). Race after technology: Abolitionist tools for the new Jim code. John Wiley & Sons.

[3] Biehler, R., Veaux, R. D., Engel, J., Kazak, S., & Frischemeier, D. (2022). Research on data science education. Statistics Education Research Journal, 21(2), Article 2. https://doi.org/10.52041/serj.v21i2.606

[4] Bowler, L., Acker, A., Jeng, W., & Chi, Y. (2017). "It lives all around us": Aspects of data literacy in teen's lives. Proceedings

of the Association for Information Science and Technology, 54(1), 27–35. https://doi.org/10.1002/pra2.2017.14505401004

[5] Bootstrap. (2022). Bootstrap:Data Science. From bootstrapworld.org.

[6] Chester, J. (2017). Empowering and protecting youth in the big data era: Issues and perspectives from the EU and U.S. Center for Digital Democracy.

[7] CODAP. (2022). CODAP - Common Online Data Analysis Platform. https://codap.concord.org/

[8] CodeHS. (2022). CodeHS data science curriculum. From codehs.com.

[9] Dasgupta, S. (2016). Children as data scientists: Explorations in creating, thinking, and learning with data [Thesis, Massachusetts Institute of Technology]. https://dspace.mit.edu/handle/1721.1/107580

[10] Data Science for Everyone. (2022). Teaching data science. K12data. https://www.datascience4everyone.org/teach-data-science

[11] Deahl, E. (2014). Better the data you know: Developing youth data literacy in schools and informal learning environments (SSRN Scholarly Paper No. 2445621). https://doi.org/10.2139/ssrn.2445621

[12] Donoghue, T., Voytek, B., & Ellis, S. E. (2021). Teaching creative and practical data science at scale. Journal of Statistics and Data Science Education, 29(sup1), S27–S39. https://doi.org/10.1080/10691898.2020.1860725

[13] EduBlocks. (2022). EduBlocks. https://edublocks.org/

[14] Finzer, W., Erickson, T., Swenson, K., & Litwin, M. (2007). On getting more and better data into the classroom. Technol. Innovat. Stat. Educ., 1. https://doi.org/10.5070/T511000025

[15] Franklin, C., & Bargagliotti, A. (2020). Introducing GAISE II: A guideline for precollege statistics and data science education. Harvard Data Science Review, 2(4). https://doi.org/10.1162/99608f92.246107bb

[16] Franklin, D., Coenraad, M., Palmer, J., Eatinger, D., Zipp, A., Anaya, M., White, M., Pham, H., Gökdemir, O., & Weintrop, D. (2020). An analysis of Use-Modify-Create pedagogical approach's success in balancing structure and student agency. Proceedings of the 2020 ACM Conference on International Computing Education Research, 14–24. https://doi.org/10.1145/3372782.3406256

[17] Gould, R. (2021). Toward data-scientific thinking. Teaching Statistics, 43(S1), Article S1. https://doi.org/10.1111/test.12267

[18] IDSSP Curriculum Team. (2019). Curriculum frameworks for introductory data science.

[19] Hautea, S., Dasgupta, S., & Hill, B. M. (2017). Youth perspectives on critical data literacies. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, 919–930. https://doi.org/10.1145/3025453.3025823

[20] Kim, B., & Henke, G. (2021). Easy-to-use cloud computing for teaching data science. Journal of Statistics and Data Science Education, 29(sup1), S103–S111. https://doi.org/10.1080/10691898.2020.1860726

[21] Krishnamurthi, S., Schanzer, E., Politz, J. G., Lerner, B. S., Fisler, K., & Dooman, S. (2020). Data science as a route to AI for middle- and high-school students (arXiv:2005.01794). arXiv. http://arxiv.org/abs/2005.01794

[22] LaMar, T., & Boaler, J. (2021). The importance and emergence of K-12 data science. Phi Delta Kappan, 103(1), Article 1. https://doi.org/10.1177/00317217211043627

[23] Lee, I., & Perret, B. (2022). Preparing high school teachers to integrate AI methods into STEM classrooms. Proceedings of the AAAI Conference on Artificial Intelligence, 36(11), 12783–12791. https://doi.org/10.1609/aaai.v36i11.21557

[24] Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. ACM Inroads, 2(1), 32–37. https://doi.org/10.1145/1929887.1929902

[25] Lee, V.R., & Delaney, V. (2022). Identifying the content, lesson structure, and data use within pre-collegiate data science curricula. Journal of Science Education and Technology, 31(1), 81-98. https://doi.org/10.1007/s10956-021-09932-1

[26] Lee, V. R., & Wilkerson, M. H. (2018). Data use by middle and secondary students in the digital age: A status report and future prospects (p. 43). National Academy of Sciences Engineering, and Medicine, Board on Science Education, Committee on Science Investigations and Engineering Design for Grades 6-12.

[27] Lee, V. R., Wilkerson, M. H., & Lanouette, K. (2021). A call for a humanistic stance toward K–12 data science education. Educational Researcher, 50(9), 664–672. https://doi.org/10.3102/0013189X211048810

[28] Martinez, W., & LaLonde, D. (2020). Data science for everyone starts in kindergarten: Strategies and initiatives from the american statistical association. Harvard Data Science Review, 2(3). https://doi.org/10.1162/99608f92.7a9f2f4d

[29] Mike, K., Hazan, T., & Hazzan, O. (2020). Equalizing data science curriculum for computer science pupils. Koli Calling '20: Proceedings of the 20th Koli Calling International Conference on Computing Education Research, 1–5. https://doi.org/10.1145/3428029.342804

[30] Papert, S. (1980). Children, computers, and powerful ideas. Harvester Press (United Kingdom). DOI, 10, 978-3.

[31] Pimentel, D. R. (2022). Tools to support data analysis and data science in k-12 education. 22.

[32] Pournaras, E. (2017). Cross-disciplinary higher education of data science – beyond the computer science student. Data Science, 1(1–2), 101–117. https://doi.org/10.3233/DS-170005

[33] Roschelle, J., Kaput, J., & Stroup, W. (2000). SimCalc: Accelerating students' engagement with the mathematics of change. Learning the Sciences of the 21st Century: Research, Design, and Implementing Advanced Technology Learning Environments, 47–75.

[34] Rosenberg, J. M., Lawson, M., Anderson, D. J., Jones, R. S., & Rutherford, T. (2020). Making data science count in and for education. In Research Methods in Learning Design and Technology (pp. 94–110). Routledge.

[35] Schanzer, E., Pfenning, N., Denny, F., Dooman, S., Politz, J. G., Lerner, B. S., Fisler, K., & Krishnamurthi, S. (2022). Integrated data science for secondary schools: Design and assessment of a curriculum. Proceedings of the 53rd ACM Technical Symposium on Computer Science Education, 22–28. https://doi.org/10.1145/3478431.3499311

[36] Schwab-McCoy, A., Baker, C. M., & Gasper, R. E. (2021). Data science in 2020: Computing, curricula, and challenges for the next

10 years. Journal of Statistics and Data Science Education, 29(sup1), S40–S50. https://doi.org/10.1080/10691898.2020.1851159

[37] Stornaiuolo, A. (2020). Authoring data stories in a media makerspace: Adolescents developing critical data literacies. Journal of the learning sciences, 29(1), 81-103.

[38] TinkerPlots: Dynamic data exploration. (2023). https://www.tinkerplots.com/

[39] Weiland, T., & Engledowl, C. (2022). Transforming curriculum and building capacity in K–12 data science education. Harvard Data Science Review, 4(4). https://doi.org/10.1162/99608f92.7fea779a

[40] Weintrop, D. (2019). Block-based programming in computer science education. Communications of the ACM, 62(8), 22-25.

[41] Wilensky, U. (1999). NetLogo (and NetLogo user manual). Center for Connected Learning and Computer-Based Modeling, Northwestern University. http://ccl.northwestern.edu/netlogo

[42] YouCubed. (2022). Explorations in data science. YouCubed High School Data Science Course. https://hsdatascience.youcubed.org/

[43] Zhang, Y., Zhang, T., Jia, Y., Sun, J., Xu, F., & Xu, W. (2017). DataLab: Introducing software engineering thinking into data science education at scale. 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET), 47–56. https://doi.org/10.1109/ICSE-SEET.2017.7