BEHAVIOUR & INFORMATION TECHNOLOGY
https://doi.org/10.1080/0144929X.2023.2295956

Taylor & Francis
Taylor & Francis Group

Preparing students to meet their data: an evaluation of K-12 data science tools

Rotem Israel-Fishelson, Peter F. Moon, Rachel Tabak and David Weintrop

University of Maryland, College Park, USA

ABSTRACT

Data science education has gained momentum in recent years. Along with the development of curricula to teach data science, the number and diversity of tools for introducing data science to learners are also multiplying. The tools used to teach data science play a central role in shaping the learning experience. Therefore, it is important to carefully choose which tools to use to introduce learners to data science. This article presents a systematic analysis of 30 data science tools that are, or designed to be, used in introductory data science education for K-12 students. The identified tools list includes spreadsheets, visual analysis tools, and scripting environments. For each tool, we examine facets of its capabilities, interactions, educational support, and accessibility. For block-based programming tools, we also examine the data science functionalities available in that tool's blocks. This paper advances our understanding of the current state of introductory data science environments and highlights opportunities for creating new tools to better prepare learners to navigate the data-rich world surrounding them.

ARTICLE HISTORY
Received 4 September 2023
Accepted 11 December 2023

KEYWORDS
Data science; block-based programming; K-12 education

1. Introduction

Over the past decade, the field of data science has experienced tremendous growth due to the exponential increase of data, enhanced computational capabilities, and growing demand for data-driven tools, processes, and algorithms. The proliferation and development of data science tools have played a crucial role in this growth, enabling data scientists to analyze and interpret large, complex datasets more effectively and efficiently than ever before (Wimmer et al. 2016). These tools have a wide range of capabilities, from simple data cleaning and wrangling tasks to complex machine-learning algorithms that can identify patterns and anomalies in vast datasets and even predict user behaviour (Haq et al. 2020).

In recent years, ideas and skills from data science have begun to enter K-12 classrooms. This push has come from educators seeking to equip students with the skills and knowledge they need to excel in an increasingly data-driven world and better understand the world around them. To that end, various data analysis tools have been developed to support students in learning about and engaging with data (Krishnamurthi et al. 2020). Additionally, existing tools that were originally designed for programming or statistics have been adapted or extended to support data science practices (Haq et al. 2020). These tools have been incorporated into new curricula designed to introduce students to data science and teach them foundational concepts and computational practices for collecting, sorting, extracting, and analyzing data from different sources (Gould 2021; Moon et al. 2023). The various analysis and visualisation tools allow students to investigate phenomena and deal with questions emerging from diverse datasets (Schanzer et al. 2022). Furthermore, such tools enable students to identify patterns and trends, extract insights from data, and make informed decisions (Ridgway 2016). To maximise the potential of data science tools in K-12 education, it is crucial for educators to carefully select the right tools for their students, as the tools themselves play a critical role in shaping learners' experiences (Pimentel, Horton, and Wilkerson 2022).

There are various aspects to consider when choosing the right tool to introduce data science to K-12 students, including the tool's functionality, the user interface, the support materials available to teachers and students, and the ease with which novices can get started. The proliferation of different tools for data analysis and the emergence of simple user interfaces for the automation of complex tasks invites students to engage in complex data science practices (Deahl 2014). However, no consensus exists on the ideal tool for introducing data science in K-12 education (McNamara 2019).

This paper builds on previous research characterising different attributes of data science tools (e.g. McNamara 2019; Wimmer et al. 2016) but focuses specifically on

CONTACT Rotem Israel-Fishelson rotemisf@umd.edu 3942 Campus Dr, College Park, MD 20742

tools used in K-12 educational contexts. Pimentel, Horton, and Wilkerson (2022) examined several spreadsheets, visual tools, and scripting tools used in K-12 data analysis across nine features, including accessibility, flexible plot creation, and extensibility. Our analysis draws on this and the preceding framework (McNamara 2019) while adding additional analytic dimensions, including statistical and graphing capabilities. This work also considers the most recent set of data science tools that coincide with the emergence of K-12 data science curricula (V. R. Lee, Wilkerson, and Lanouette 2021). After compiling a set of educational data science tools, we examined each tool to answer the following research questions:

For the current set of data science tools used in K-12 classrooms:

RQ1: What capabilities does the tool possess regarding data science practices, data visualisation, statistical calculations, and extensibility?
RQ2: How do users interact with the tool, and how might the interaction support learning?
RQ3: What accessibility features does the tool provide?
RQ4: If the tool uses a block-based programming approach, what types of blocks does the tool include to support students learning data science?

2 Prior research

The tools offered for K-12 data science education include diverse capabilities for collecting, analyzing, and visualising data inside and outside the school walls (Deahl 2014). They range from simple spreadsheets to complex scripting languages like Python and R. These tools differ in their ease of use, learning curve, user interface, and functionality. Pimentel, Horton, and Wilkerson (2022), in their analysis of educational data science tools, suggest dividing the tools into four broad categories: spreadsheets; visual interfaces; scripting languages; and other tools.

Spreadsheet tools, such as Microsoft Excel and Google Sheets, are commonly used in data science education because they are ubiquitous, easy to use, and freely available for educational purposes. Spreadsheets enable the collection of data from various sources. They can be used to clean, organise, filter, manipulate, and prepare data for analysis. Moreover, spreadsheets support data exploration and analysis. They can be used to perform both basic and advanced statistical analyses. Furthermore, spreadsheets offer visualisation capabilities by creating charts, graphs, and other visualisations. These capabilities can help students understand statistical concepts, identify trends and patterns, and interpret the data (Bargagliotti et al. 2020; V. R. Lee, Wilkerson, and Lanouette 2021). Nevertheless, spreadsheets are limited in their ability to perform sophisticated analysis, and thus, it is often required to load them into other data science tools (Gavriilidis et al. 2023).

Visualisation tools provide graphical user interfaces for building visual representations of data. Their interfaces often include menus or drag-and-drop features to ease the analysis and understanding of complex data. In addition, these tools provide functionality for arranging quantitative and qualitative data in tables, graphs, and other graphic depictions so that users can discover trends in a dataset without knowing how to programme (Deahl 2014). The interactive, friendly user interface also supports transforming data representations and performing exploratory analyses from various perspectives (Pimentel, Horton, and Wilkerson 2022). Under this category are dedicated tools developed for educational purposes, such as CODAP (2022), Tuva (Erickson 2016), and DataClassroom (2022). These tools constitute a scaffold for learning data science practices, and their learning curve is small. However, they have limited analytical capabilities and the ability to manipulate raw data. There are also professional visualisation tools, such as Tableau (2023) and Power BI (Microsoft 2023) which offer advanced capabilities and enable the creation of dashboards and interactive reports (Batt et al. 2020).

Scripting languages include R, Python, and Pyret, and they are used to perform data science activities and manipulations, statistical analysis, and machine learning. These languages offer the widest functionality but often have a steep learning curve (Haq et al. 2020). These languages are used in educational settings despite being more complicated than other analytic tools because they enable the automation of advanced functions on large datasets (LaMar and Boaler 2021; Pimentel, Horton, and Wilkerson 2022). Python is a widely used general-purpose programming language that has a large community of users and various libraries and modules that make working with data easy. Similarly, R is another popular data science scripting language designed for statistical analysis and data visualisation. Pyret is a functional-programming language with Python-like syntax designed for educational purposes, featuring colour-coded error messages written in student-friendly language.

A recent study of 330 teachers from across the US found that the most common tool is spreadsheets, while they used tools more common in professional practice less often. Among the barriers to using the tools were: the necessary resources (the cost of the

tool and time to develop dedicated lessons) and the required learning curve (Rosenberg et al. 2022).

Increasingly popular in educational environments are block-based programming tools, which serve as alternatives to scripting languages like R and Python, and make the act of programming more accessible and less error-prone for novices (Weintrop 2019). A recent survey of block-based programming tools identified over 100 unique environments with various capabilities, including numerous tools with support for data science activities (Lin and Weintrop 2021). By using block-based programming tools, students can avoid the frustration of learning the syntax of a programming language. Instead, they can focus on developing the computational core ideas (Bau et al. 2017). The use of block-based programming in K-12 classrooms can lead to better learning outcomes than students who learn using text-based languages (Weintrop 2021). Several block-based programming tools have been built specifically for data science, e.g. BlockPy (Bart et al. 2015) and PlayData (Fernandez, De Deus Lopes, and Blikstein 2023) and with many others having some data science functionalities while not being designed specifically for this purpose (Kross and Guo 2019; Lin and Weintrop 2021).

Other tools include environments that support scripting along with scaffolded workbooks. These environments offer built-in interactive exercises for performing step-by-step operations of changing existing code or writing code incrementally (Pimentel, Horton, and Wilkerson 2022). One such tool is Jupyter Notebook, an open-source web application that allows users to run chunks of Python code and immediately display output and text notes in the same document (Chandel et al. 2022). Google Colab (Bisong 2019) is another tool that falls under this category.

3. Methods

To answer the stated research questions, we conducted a systematic analysis to identify the current set of tools and environment that are being used in introductory K-12 data science education. Our first step in assembling the list of tools was to review tools presented and discussed in the academic literature. The analysis began by searching the keywords 'Data Science Education Tools' in the Association for Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers (IEEE) digital libraries. Next, we examined the articles' title, abstract, and methodology to identify whether a data science tool was included as part of the research. Next, we added to the list additional tools recommended by colleagues not identified in the systematic search. This process yielded a set of 30 tools. After compiling a list of data science education tools, we created inclusion and exclusion criteria for selecting the relevant tools for analysis. To be included in this analysis, the tool: (1) must be used for learning or focused on education; (2) must be focused on data science concepts or practices; (3) must be available for analysis (i.e. can be run and used). Exclusion criteria were: (1) must not be focused on AI or machine learning; (2) must not be a library (e.g. pandas) or general-purpose programming language (e.g. R, Python).

To answer our stated research questions, we identified several distinct analytic dimensions for each of our four research questions. For the first three research questions, the dimensions were informed by prior work evaluating data science learning environments (McNamara 2019; Pimentel, Horton, and Wilkerson 2022). Table 1 presents the 15 analytic dimensions organised by guiding research questions:

To answer our fourth research question focused on block-based programming tools, we drew on Lee and Delaney's (2022) categories for topics in data science education and further supplemented the list with topics included in GAISE-II, a standards document for statistics education (Franklin and Bargagliotti 2020), and the International Data Science in Schools Project (IDSSP) Framework, a standards document for data science education (IDSSP Curriculum Team 2019). The results was a list of 14 topics that are, or should be, taught in K-12 data science. Table 2 outlines the 14 categories generated through this combination. Having compiled this list of topics, we analyzed every block-based environment to see if it had blocks associated with the data science topic.

Having defined the analytic dimensions for each data science education tool. Two researchers analyzed each environment independently and recorded their results in a spreadsheet. Agreement in the first round of coding was 89.9%, and in the second round (after functionalities were further discussed) was 95.3%. All discrepancies were discussed with the full research team until agreement was reached.

4. Findings

Table 3 presents the list of the 30 data science tools used in K-12 data science education that were analyzed for this work.

4.1. RQ1: K-12 data science education tool capabilities

To answer our first research question, which focused on the capabilities of the tools, we identified five distinct

Table 1. Analytic dimensions for the first three research questions.

RQ	Dimensions	Description
RQ1 Capabilities	Data Manipulation	The processes for organising and extracting data, and how they are carried out.
	Statistical Capabilities	Can the tools be used to conduct correlation analysis and linear regression?
	Data Visualisation	Support for tabular and graphic displays of the data, the type of graph creations supported by the tool, and their creation method.
	Dataset Availability	The existence of built-in datasets, how these datasets are accessed through the tool, what topics these datasets cover, and capabilities to import data (as a file or via API calls from external services) and export the data.
	Extensibility	Open-source usage and flexibility to build extensions.
RQ2 Interactions & Education	Runtime Environment	The environment in which the tool is executed, (web browser, or operating system)
	Programming Modality	The representative infrastructure used and the variety of interactions that the interface enables.
	Scripting Language	The programming language is supported by the tool.
	Instructional Materials	The tool includes resources aimed to assist educators and students in their teaching and learning processes, including written guides, tutorials, lesson plans, etc.
	Activities/ Assignments	The ability for instructors to create and/or assign activities for their students within the tool.
RQ3 Accessibility	Screen Reader	Is the environment compatible with screen reader software?
	Colour Palette	Does the tool consider colour blindness or offer accessible colour palettes?
	Simplified Representations	Does the tool provide the ability to simplify the provided graphical data representations?
	Customised Interactions	Can the user customise interactive features to support learners with limited mobility or fine motor skills?
	Font size Modification	Can the user customise the font size to support those with limited sight?

analytic categories (Table 1). A comprehensive listing of each tool and how it was scored for each dimension can be found in Appendix 1.

4.1.1. Data manipulation

To begin our analysis of K-12 data science education tool capabilities, we investigated the data manipulation capabilities built into each tool. This is important as the learner's ability to transform data is essential for meaningful analysis and interpretation. Twenty-six of the 30 tools we reviewed featured some sort of data manipulation capability. Our analysis of the tools revealed that the data manipulation capabilities featured most prominently were filtering, deleting, sorting, and aggregating. Our analysis highlights the prevalence of each of these, with consideration of how data manipulation occurred. Only one tool (of the 26 that included data manipulation capabilities) did not support filtering of data. Sorting (available in 20 tools), deleting (available in 18 tools), and aggregating (available in 18 tools) were less prevalent. Fourteen of the 30 tools we reviewed included all four data manipulation capabilities. Of these 14 tools, 9 of them utilised a graphical user interface (GUI) for data manipulation (CODAP, DataClassroom, DBSnap, EduBlocks, JASP, MakeCode, PlayData, Tableau, and TinkerPlots), while 4 (BridgesCS, Google Colab, Jupyter Notebook, and Quorum) utilised code-based data manipulation. One of these 10 tools (Kaggle) used both GUI and code for data manipulation.

4.1.2. Statistical capabilities

Our second analytic dimension is the statistical capabilities of each tool. We evaluated if the tools could conduct

Table 2. K-12 data science topics used to analyze block-base data science educational tools.

RQ	Data science topic	Description
RQ4 Data Science Block Features	Ethics	Considering privacy and bias
	Inquiry with Data	Asking different kinds of questions; the statistical investigation cycle
	Distributions & Variability	Distributional shape, normal distribution, standard deviation and other measures of spread
	Measures of Center	Mean, median, and mode
	Programming – Commenting	Ability to 'comment' within a section of block-based code
	Programming – Logic	Boolean logic and conditional statements
	Programming – Functions	Ability for the user to define custom algorithms and callable chunks of code
	Transformation	Ability to transform data programmatically
	Variable Association	Linear regression, correlation, and statistical tests
	Graphs & Figures	Ability to create graphs, plots, and figures to visualise data
	Sampling/Simulation	Ability to draw a randomly-generated sample from the total data loaded, and/or to repeat this process multiple times for simulation purposes
	Machine Learning	Support for machine-learning models (beyond simple linear regression); decision trees, k-means clustering
	Time Series	Blocks specifically for managing time series data
	Map Data	Blocks specifically for managing location and map data

Table 3. List of the 30 tools that were identified and analyzed.

Tools	Brief Description	Academic References
Blockly	Blockly is an open-source block-based programming library.	(Fraser 2015; Weintrop et al. 2017)
Blockly-DS	Blockly-DS is a data science extension to the Blockly visual programming language tool.	(Barboza et al. 2023)
BlocklySQL	BlocklySQL is a block-based editor for SQL.	(Pöhner et al. 2019)
BlockPy	BlockPy is a web-based, block-based Python environment designed for data science.	(Bart et al. 2015; 2017; Poole 2017)
Bridges CS	Bridges CS is an easy-to-use interface with a set of classes (C++ and Java are supported) that serve as building blocks to the common data structures (lists, tree structures, graphs) used in computer science.	(Strahler et al. 2020)
CODAP	CODAP is a free educational software for data analysis.	(Frischemeier et al. 2021; Mojica, Azmy, and Lee 2019)
DataClassroom	DataClassroom is a web-based data analysis and visualisation application.	(Rosenberg et al. 2022)
Datacommons	Datacommons provides a visualisation tool to explore aggregated data from different sources.	(Sheth, Padhee, and Gyrard 2019)
Dataland	Dataland is a new block-based programming system designed to help middle- and high-school-aged students learn and do data analysis and visualisation.	(Wang and Dasgupta 2022)
DBSnap	DBSnap is a web-based application to build database queries with block-based tools.	(Silva and Chon 2015)
EduBlocks	EduBlocks is a visual block-based programming tool that helps teachers to introduce text-based programming languages, like Python & HTML, via a drag-and-drop programming experience.	(Lindín, Steffens, and Bartolomé 2022)
GapMinder	Gapminder identifies systematic misconceptions about important global trends and proportions and uses reliable data to develop easy-to-understand teaching materials to rid people of their misconceptions.	(Le 2013)
Google Colab	Google Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis, and education.	(Bisong 2019; Nelson and Hoover 2020; Tock 2019)
GP	GP is a general-purpose block-based language that enables generating high-quality graphics, manipulating images and sounds, and analyzing text files or CSV datasets.	(Kelleher et al. 2017)
iNZight	iNZight is a free, easy to use software for statistical data analysis.	(Wild 2018; Wild, Elliott, and Sporle 2021)
iSnap	iSnap is a data-driven, intelligent tutoring system for block-based programming, offering support such as hints, feedback and curated examples.	(Marwan and Price 2023; Price et al. 2019; Price and Barnes 2017)
JASP	JASP is a free, open-source GUI-based statistics programme that offers both classical and Bayesian analysis.	(Hersh et al. 2023; Love et al. 2019)
Jupyter Notebook	Jupyter Notebook is a 'code notebook' that allows users to run chunks of Python code, along with displayed output and text notes all in the same document.	(Werner et al. 2021)
Kaggle	Kaggle is an online community platform for data scientists and machine learning. It allows users to collaborate with other users, find and publish datasets, and use GPU integrated notebooks.	(Bojer and Meldgaard 2021; Hoque, Coelho, and Mueller 2019)
MakeCode Data Science Editor	An experimental block-based editor to teach data science to middle school students.	(Meitiner and Seneviratne 2020)
mBlock	mBlock is a graphical programming platform tailored to coding education. It allows users to switch between block-based interface and programming interfaced based on Python.	(Peng, Bai, and Siswanto 2020)
NetsBlox	NetsBlox is a cloud-based block-based tool that enables the creation of networked programmes.	(Brady et al. 2022; Broll et al. 2017)
PlayData	PlayData is an extension for Scratch including several blocks for data analysis.	(Fernandez, De Deus Lopes, and Blikstein 2023)
Pyret	Pyret is a programming language and environment designed for teaching computer science to beginners.	(Politz et al. 2018)
Quorum	An evidence-oriented programming language tool that supports blind or visually impaired students.	(Ladner and Stefik 2017; Stefik and Ladner 2017)
Scratch Data Blocks	Scratch Data Blocks is a Scratch-based toolkit that allows Scratch users to programme with public data from the Scratch online community.	(Dasgupta 2015; Dasgupta and Hill 2017)
Snap!	Snap! is a block-based programming language and platform for creating programmes.	(Feng, Tilevich, and Feng 2015; Harvey et al. 2014)
Tableau	Tableau is a leading data visualisation tool used for data analysis and business intelligence.	(Geise 2021; Khalajzadeh et al. 2022)
TinkerPlots	TinkerPlots is a data visualisation and modelling tool.	(Fitzallen 2020; Saldanha and Hatfield 2021; Watson and Donne 2009)
Tuva	Tuva is a platform that enables students to easily explore, manipulate, and analyze data.	(Pimentel, Horton, and Wilkerson 2022; Reiten and Strachota 2016)

correlation analysis and linear regression, given that these are the two most commonly used techniques for investigating quantitative relationships. Sixteen of the tools we reviewed offered both correlation and regression analysis. Twelve of the 14 tools that had the full range of data manipulation capabilities also supported correlation and regression analysis; the exceptions were DBSnap and PlayData, which did not support these capabilities. There were also 5 tools that had limited data manipulation capabilities, yet still offered correlation and regression analysis: BlocklyDS, iNZight, Pyret, TinkerPlots, and Tuva. In addition to this summary, we noted that several tools possess more advanced statistical capabilities. BlocklyDS, for example, allows users to create decision trees, and JASP supports Bayesian versions of common analyses

and machine learning applications. Similarly, Tableau supports more complex modelling than simple correlation and linear regression such as clustering and predictive analysis.

4.1.3. Data visualization

As part of our analysis, we examined which data visualisations the tools support. Specifically, we focused on tabular displays, the types of graphs supported, and how users create them. Most of the tools (21 out of 30) allow the presentation of the data in tabular format. All but one (BlocklySQL) allow visualising the data in a graph. Seven tools enable the creation of graphs by writing textual code, and ten by writing short block-based programmes (Figure 1, left). We also found three tools, BlockPy, EduBlocks, and mBlock, which support both methods and allow users to seamlessly alternate between the two creation methods. Additionally, we found that nine tools enable the creation of graphs using a graphical user interface (GUI). In these tools, the user can create a graph by dragging and dropping attributes into the axes of a predefined visualisation (Figure 1, right), often in the form of a scatterplot, as found in 27 tools. Other types of graphs that were supported by the tools are line graphs (21 tools), bar charts (18 tools), pie charts (14 tools), box plots (12 tools), and histograms (16 tools). Beyond these basic visualisation approaches, several tools support the creation of more sophisticated charts like radar graphs (in EduBlocks), violin charts (Jupyter notebook and Quorum), and heatmaps (MakeCode Data Science Editor).

4.1.4. Data availability

Since data science is fundamentally about data, it was important to consider how data is accessed within each tool. Our analysis shows that only 15 of 30 tools reviewed include built-in datasets, which enable students to explore provided data while learning about and practicing data science techniques. Some tools offer a limited number of built-in datasets, while others include a wide variety. BlockPy, for example, provides 60 built-in datasets which cover topics related to entertainment (i.e. data about Broadway shows and video games), environment (i.e. wind turbines, earthquakes, global emissions), health (i.e. COVID-19, hospitals), education (i.e. school scores), and politics (i.e. state demographics, elections). A review of the topics of the built-in datasets for the 15 tools that provide them is based on categories used in materials from Bootstrap: Data Science (2022) and is presented in full in Table 4 below.

Alongside providing datasets for learners, allowing learners to import data is an important feature for introductory data science tools. Of the reviewed tools, 27 allow users to import datasets, and 27 tools enable exporting all or part of the datasets. A third approach for gathering data is application programming interface (API) calls. API calls enable data retrieval from web services or external databases. Our analysis revealed that seven tools support API calls to access data from external services. NetsBlox, for example, allows users to query and retrieve data from different services on topics including maps, weather, earthquakes, movies, games, museums, social media, and more. The data from these services are accessed using an API call integrated into the user interface in the form of dedicated blocks with dropdown lists of the services and arguments (Brady et al. 2022).

4.1.5 Extensibility

Extensibility is an essential aspect of data science tools because it allows the flexibility to adapt to evolving computational methods and analysis requirements of different users. Moreover, the ability to build extensions based on existing modules prevents them from becoming obsolete (McNamara 2019). Therefore, we examined whether each data science tool supported the ability to develop extensions and whether it has an open-source policy that allows customisation. We found eleven open-source tools that allow the creation of tool extensions. Jupyter Notebook, for example, has several extensions that allow users to customise their notebooks and

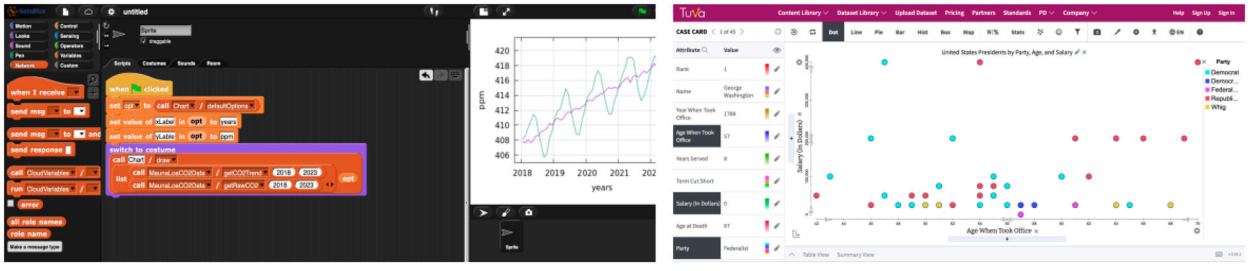

Figure 1. Creation of graph using NetsBlox's inherent blocks (https://netsblox.org/) (left) and creation of graph using Tuva's graphical user interface (https://tuvalabs.com/) (right).

Table 4. Topics represented in built-in datasets.

Tool	Entertainment	Sports	Politics	Environment & Health	Education
BlocklyPy	✓	✓	✓	✓	✓
Bridges CS	✓			✓	
CODAP	✓		✓	✓	✓
DataClassroom				✓	
DBSnap					✓
GapMinder			✓		
iNZight			✓		
JASP	✓	✓	✓	✓	✓
Jupyter Notebook	✓		✓	✓	✓
Kaggle	✓	✓	✓	✓	✓
NetsBlox	✓	✓	✓	✓	✓
Quorum	✓	✓	✓	✓	
Tableau	✓	✓	✓	✓	✓
TinkerPlots	✓	✓	✓	✓	✓
Tuva	✓	✓	✓	✓	

add additional functionality, such as table of contents and variable inspector. Blockly enables users to enhance its functionality by creating custom blocks using the Blockly extension builder without writing code or developing new blocks using JavaScript. A notable advantage of open-source policies is that they allow changes to the code to become public domain at no cost (Wimmer et al. 2016). The developed extensions will sometimes be published or even embedded as an integral part of the tool. In CODAP, for example, a Plugins section was added to present various extensions developed by users. One such extension is called 'transformers', which enable transforming datasets to produce new, distinct output or values instead of modifying the original input dataset. The Transformers plugin allows users to compare variables, perform advanced filtering and aggregation, and run statistical analyses (The Brown PLT Blog 2021).

4.2. RQ2: K-12 data science tool supported interactions and educational features

To answer our second research question, we first examined how learners engage with the tools and which educational features and supports they provide. A complete listing of each tool's interactions and educational features can be found in Appendix 2.

4.2.1. Runtime environment

Our analysis revealed that most tools (28 out of 30) can be reached and used through a web browser. Of these, eight tools (GapMinder, GP, iNZight, JASP, Jupyter Notebook, mBlock, Quorum, and Tableau) also enable local installation of their environment using the Windows or Mac operating systems. Two additional tools, Bridges CS and TinkerPlots, can only be accessed by installing them on Windows or Mac. mBlock provides the broadest access among the tools as it can be run on a browser, Windows, Mac, Linux, or Android operating systems. The runtime environment is an essential parameter because there is no uniformity among the operating systems students use. Providing a browser-based approach makes it easier for learners to access the tool regardless of the underlying hardware and also makes it easier for the developer to update the tool without it needing to be reinstalled or manually updated. However, tools that run in a browser require a constant and stable internet connection, which may not always be available. In contrast, local installation requires technological literacy from the student and appropriate administrative privileges on the machine.

4.2.2. Programming modality

The data science tools analyzed differed in their interface design. Eight tools offer a user interface that allows manipulating data and creating visualisations by selecting from menus or drag-and-drop components. The rest of the tools offer block-based, text-based, or hybrid interfaces, which we refer to as programming modality. Programming modality encompasses the representational infrastructure used and the range of interactions facilitated by the interface (Weintrop and Wilensky 2018). Block-based interfaces involve dragging and dropping graphical blocks and snapping them together to form a script. Such modality may limit the ability to write complex programmes, but is visually intuitive and helps prevent syntax errors. We found 15 tools offer a block-based interface. An example of such an interface can be found in Snap!, where users can interact with the programme handling the data in a block-based form (Figure 2).

Text-based interfaces involve writing code using a text editor. This modality provides greater flexibility, more concise expressiveness, and often more fine-grained control over the programme's behaviour. However, it can be difficult for novices to learn due to the

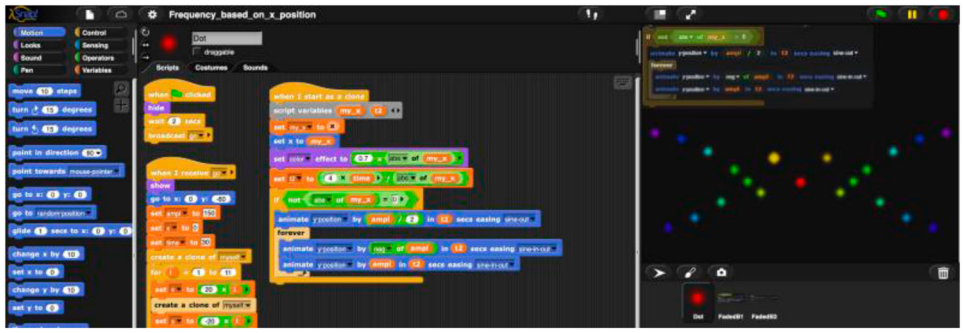

Figure 2. Snap! (https://snap.berkeley.edu/).

syntax and structure required to write the code. We found eight tools that incorporate text-based programming. In Google Colab and Kaggle, for example, users can assemble their code in either Python or R languages, while in Bridge CS, they can choose between C++, PHP, and Python (Figure 3).

Hybrid interfaces combine elements of both block-based and text-based interfaces. This modality can provide a smoother learning curve for novices and help bridge the gap between visual and traditional text-based programming (Weintrop and Holbert 2017). Using the categories defined in Lin and Weintrop (2021), we further break these tools into sub-categories based on the interaction and presentation of the modalities. Read-only one-way transitions, like Blockly, DBSnap, and mBlock, enable viewing the text-based version of the block-based programme. Editable one-way transition tools, like EduBlocks, allows viewing the blocks' textual version and even editing them but do not support text-based edits being converted back to blocks. Dual-modality tools, such as BlockPy and BlockSQL, enable switching between the block and the text-based representations where a change in one modality affects the other.

4.2.3. Scripting language

We examined which scripting/programming languages are used by the tools analyzed in this study. Our analysis shows that the most common scripting language is Python, used by Blockly, BlockPy, Bridges CS, EduBlocks, Google Colab, Jupyter Notebook, Kaggle, and mBlock. The next most common languages are R (DataClassroom, Google Colab, Kaggle) and SQL (BlockSQL, DBSnap). Blockly supports the widest variety of scripting languages and provides a drop-down menu to quickly switch between code views in multiple languages.

4.2.4. Instructional materials

Most of the tools (20 out of 30) provide instructional materials. These materials include written guides, video tutorials and demos, presentations, lesson plans, and sample projects. Some tools even have a section or repository dedicated to instructional materials. In DataClassroom, for example, there is a user guide detailing how to work with data, conduct statistical analysis, and create visualisations. Additionally, under their resource library, they have ready-to-teach datasets accompanied by complete lesson plans. Teachers and students can use such materials to perfect their skills while working with the tools. Indeed, in recent research with math, computer science, and data science instructors, highlighted the importance of such instructional resources for bringing data science into the classroom (Schwab-McCoy, Baker, and Gasper 2021).

4.2.5. Classroom assignments creation

Along with instructional materials, we also examined whether the tools enable the creation of learning activities in an integrated manner. Seven of the 30 studied

Figure 3. Google Colab (https://colab.research.google.com/).

tools allow teachers to create classroom assignments. For example, in EduBlocks, instructors can create an assignment that includes a name, description, and initial project created using the tool, and then can link it to a specific class and then grade it after submission. Other tools provide more basic features that can be used to write assignments for learners. BlockPy, for example, has a component for creating task instructions with a built-in text editor that allows adding links, images, and tables. DataClassroom and CODAP enable embedding GoogleDoc or HTML files to create a guided assignment. mBlock enables the creation of assignments using the integration of Google Classroom.

4.3. RQ3: K-12 data science education tools accessibility features

The third research question attends to the accessibility of each tool, analyzing them for five kinds of accessibility features based on prior work on software accessibility (Marriott et al. 2021). A complete listing of each tool's accessibility features can be found in Appendix 3.

4.3.1. Screen reader support

To evaluate screen reader capabilities, we started with a 'tab test', which means we pressed the 'tab' key repeatedly to see if relevant objects were selected. This is important as this is how a screen reader navigates a graphical user interface to the narrative of the various features and objects. Only two of the tools analyzed in this study, Tuva and DataClassroom, passed this test. Part of the reason for the low success rate is due to the fact that many of the K-12 data science tools have coding or interactive graphing panes that are tabbed to as a single object, which the screen reader will often describe the tool as a 'Coding Window' or a similar term. While this tells the user where they are on the screen, it is not useful to a visually impaired learner.

4.3.2. Colour palette

To evaluate whether or not the environment included accessible colour palettes for colour-blind users, we started by creating a basic graph (ideally a pie chart or side-by-side bar chart) to compare the colour palette to a list of problematic colour combinations for people with common colour vision deficiencies (e.g. green & red, green & brown, blue & purple, etc.). Seven tools were removed from this analysis because they did not have plotting capabilities; of the remaining 23, all 23 had default colour combinations that did not rely on the problematic pairs listed above.

4.3.3. Simplified representations

Our next criteria investigated if or how it was possible to simplify the graphical representations provided or generated by the tool. This is important as removing extraneous data, or visual elements (e.g. unneeded lines and redundant labels) can be beneficial for neurodiverse learners. Again, seven tools were removed from the first analysis because they did not support plotting. Of the remaining 23, only two tools examined (GapMinder and BridgesCS) supported some options to simplify graphical representations, but the options were limited.

4.3.4. Customised interactions

Our penultimate accessibility criteria investigated if the data science tool supported customised interaction patterns. Particularly, if there was an option to customise how the users created or interacted with data graphics. This is important as such flexibility can make tools accessible for learners with fine motor difficulty. Adjusting the sensitivity of inputs or the movements associated with zooming or panning can make tools more accessible. None of the tools provided an option to customise the standard features for interacting with a graph or draggable blocks: the closest was GapMinder, which allows the learner to add the ability to pan with the cursor and zoom with the mouse wheel but does not allow you to modify the movement necessary for these actions. This feature should be strongly considered for existing and future projects in data science education, particularly for drag-and-drop block-coding or data-visualisation programmes where a motor disability might significantly hinder a student's ability to use the tool.

4.3.5. Font size modification

The final accessibility criteria examined if tools allowed the learner to modify the font size. This can be an important accessibility feature for learners with limited vision or difficulty with reading. Ten of the studied tools supported this behaviour (two using some form of 'zoom' button that magnified the block-coding pane). While this is somewhat mitigated by most web browsers allowing magnification of the current tab, effectively changing the font size of the window, we were surprised to see that most of the tools we surveyed did not include some ability to change font size within the tool.

4.4. RQ4: block functions in block-based data science tools

The fourth research question is focused specifically on block-based programming tools included in our analysis

and investigates the types and functionalities of the blocks provided. The 13 block-based tools included in this analysis are: Blockly, BlocklySQL, BlockPy, DBSnap, EduBlocks, GP, mBlock, NetsBlox, Scratch, Snap!, Playdata, Dataland, and BlocklyDS. As mentioned in the methodology section and detailed in Table 2 above, our analysis was inspired by the categories that emerged in the study of Lee and Delaney (2022), the GAISE-II standards (Franklin and Bargagliotti 2020), and the IDSSP Framework (IDSSP Curriculum Team 2019). A complete listing of data science functionality coverage in the different tools can be found in Appendix 4.

4.4.1. Data science topics most commonly found in block-based tools

Several functionalities were found in most of the block-based tools surveyed. Every tool examined, for example, included blocks for Boolean logic and conditional statements. Such blocks are very important as they control the flow of a programme and determine whether a section of code should be executed based on the evaluation of a Boolean statement (i.e. if a given statement is true or false). Ten of the 13 tools included blocks for user-defined functions (absent in BlocklySQL, DBSnap, and Dataland), 10 of 13 tools included blocks for transforming data (absent in Blockly, BlockPy, and EduBlocks).

4.4.2. Data science topics less commonly found in block-based tools

Many functionalities were not found in many of the block-based tools surveyed, including some that seem integral components of a data science tool. For example, only two tools – Blockly and BlocklyDS – had a block for calculating measures of variability, such as standard deviation. Similarly, only four tools – Blockly, BlocklySQL, PlayData, and BlocklyDS – had a block for calculating basic descriptive statistics, such as mean, median, and mode. Most sophisticated statistical tests, such as correlation and linear regressions were less common – only BlocklyDS had blocks to support these tests. Blocks to create graphs and figures were also rarely found, as only four tools (BlockPy, EduBlocks, Dataland, and BlocklyDS) had blocks for this functionality. It should be noted that in some tools, this functionality does exist through user interface components rather than the blocks. For example, in DBSnap, graphs are created by clicking an icon next to the query result display and setting the properties in a dedicated pop-up window. Other rare functionalities included sampling and simulation (EduBlocks, BlocklyDS), managing map data (Dataland, Netsblox), and the ability to add a comment amidst block-based code (GP). As for map data, there is a difference in the way the blocks are presented in the interface. While in Dataland, there is a main category that gathers all the relevant blocks, in NetsBlox, the blocks are hidden under services to be loaded into the interface. Notably, only one block-based tool (BlocklyDS) offered machine-learning capabilities. Specifically, the tool has different blocks for performing predictive analysis, such as logistic regression, KNN, Naïve Bayes, decision trees, neural networks, and K-Means (Barboza et al. 2023). Overall, BlocklyDS possessed the most of these less common functionality blocks, and blocks for the most of our coding categories out of the reviewed tools (10 of our 17).

4.4.3. Data science topics not found in block-based tools

Several categories from our coding scheme were not present in any of the block-based tools included in this analysis, including Ethics, Inquiry with Data, and Time Series Data. The first two of these missing categories were identified by Lee and Delaney (2022) as part of data science curricula, but are not natively supported by the blocks present. This means the topics must be taught beyond the tools itself (e.g. a lesson on Data Ethics taught without the tool being the focus of engagement). This highlights the importance of the curricula that accompany the tools and serves as a reminder that the tool on its own might not be sufficient for covering all data science topics.. The last absent category, Time Series Data, means none of the block-based tools provide capabilities for analyzing or visualising time series data and suggests a potential direction for future expansion of the current set of data science tools.

5. Discussion

This paper presents a systematic analysis of data science tools that are being used, or designed to be used, in introductory data science education. Our analysis identified 30 tools, which we then analyzed according to 29 dimensions, organised around four areas to deepen our understanding of the current state of tools designed to introduce novices to data science. Such tools are essential enablers for data science education, and it is thus crucial to understand the affordances, drawbacks, and differences between available tools. In this section, we reflect on the results of our analysis and highlight encouraging trends, opportunities for future design work, and the implications of these findings for both designers and educators.

5.1. Trends in data science tools for K-12 learners

In looking across the analytic dimensions employed in this work, we see some emerging trends and interesting tensions in the design of K-12 data science education. One encouraging pattern across the tools is the ease with which they allow exploring exciting and diverse datasets. This can be seen by the tools providing built-in datasets and supporting mechanisms to import new data sets. Our review of built-in datasets within those tools that include them indicates that they generally include a wide array of topics, as most of the tools included at least one dataset on Entertainment, Sports, Politics, Environment & Health, and Education. Less than a third of the tools surveyed included built-in datasets; however, all but 3 tools reviewed allowed for the import of data from an external source. This ability alleviates the limited number of tools with built-in data by allowing for a theoretically infinitely flexible supply of data to analyze in the rest of the tools, albeit external sources that need to be identified and loaded in by the user.

One particular approach that we think is promising is scaffolded support for querying external APIs. Providing low-threshold approaches to allow novices to explore and pull in data from publicly available datasets can broaden the questions that learners can pursue and increase the authenticity of educational data science activities as learners will be learning with real, live datasets. Coupling live data with interfaces that allow novices to explore trends and answer questions they are interested in can make for a compelling introductory data science learning experience and help K-12 students understand the power and utility of having foundational data science skills.

In examining the tools identified as being used in K-12 data science education, an interesting tension emerges in the relationship between data science and programming. While some tools make programming the central mechanism by which learners will engage in data science (e.g. Jupyter Notebook, iSnap, Quorum, Kaggle) other tools do not include any programming or programming-like interfaces for learners (e.g. CODAP, GapMinder, Tuva). Further, how coding is presented to learners (i.e. graphical block-based vs. conventional text-based programming) also highlights a difference in what is prioritised by the tool and what is being foregrounded. The decision of if and how programming is being included in the environment impacts what ages/grades the tool is appropriate for, the necessary prior knowledge needed by the educator, the level of scaffolds needed by the environments, and the ways the tool can and cannot connect with data science tools and practices beyond the specific learning environment. Given that this review was interested in data science tools that span K-12, it is not surprising to find a breadth of ways of engaging (or not engaging) with programming. Thinking through how these tools might fit together and what tools fit where across the K-12 spectrum is an important open question for further work.

In our review of data science functionality in block-based programming tools, we were surprised by the absence of many functionalities we expected to be commonplace in such tools. For example, blocks to calculate the mean or standard deviation of a list are relatively uncommon in the tools surveyed. The blocks present in many of these tools are indicative of their development from programming education tools: computer science functionalities like conditional statements and user-created function blocks are quite common, while data science functionalities (such as calculating variable association or creating data visualisations) are less consistent across block-based tools. One strong exception is that most tools provide some means for the user to manipulate and transform data loaded into the tool.

The integration of data science tools and block-based environments into K-12 education shows promise in enhancing students' understanding of data science concepts. Block-based environments provide an accessible entry point, while data science tools offer authentic experiences.

5.2. Implications for designers

The analysis presented in this work has several implications for designers and suggestions for future work. The first set of implications is in response to the third research question exploring various ways tools attend to accessibility. It is critical that the designers of K-12 data science tools consider accessibility as a foundational goal. This is a legal requirement in the United States and a central aspect of creating equitable tools. In this analysis, accessibility was operationalised across five dimensions, some of which should be relatively easy to address in the design or re-design of data science tools. Giving special consideration to the colour palette used (or offered), allowing learners to modify the text size, and providing options for simplifying data visualisations seem like relatively straightforward features to implement/add and should be prioritised in the design/re-design of K-12 tools. Other accessibility features, such as supporting customised interactions and screen readers, may take more effort but are still important and worthy pursuits for designers of educational data science tools.

A second implication for designers is acknowledging the various practices and skills that constitute data science and thinking through if/how their tool supports the breadth of what it means to engage in data science. While data visualisation and statistical analysis seem to be prioritised in the reviewed tools, thinking through how to support the identification of relevant datasets, how to clean and normalise them, and how to communicate insights are all critical parts of the data science process. It should be mentioned that many tools already exist for these data science aspects. However, such tools were not included in the analysis because of the specific exclusion criteria. Our analysis suggests there is an opportunity for designing new K-12 data science tools that emphasise these other aspects of data science. In particular, a tool that helps learners identify potentially useful datasets and determine if/how a given data might be useful for answering a question is emerging as an under-explored design space in K-12 data science education.

Finally, our work has particular relevance for both curriculum designers looking to choose a block-based programming tool for their data science course, and block-based tool developers considering what blocks to include for a tool meant for data science education. We have reviewed existing block-based data science tools and identified strengths but also gaps where some expected functionalities are missing. Curriculum designers should consider the capabilities of the tool they choose to be used in their course, as some tools have the blocks to manage some aspects of data science practices better than others. Tool developers should review this list and note the gaps where a tool providing more of the missing blocks might be well-appreciated by curriculum designers, teachers, and students alike. Importantly, the absence of categories like Ethics and Inquiry with Data reminds us that there are lessons in data science that cannot be directly mapped to functions in a tool. These concepts are more in the realm of critical thinking on the part of the student, and curriculum designers should ensure they are well-covered in a data science course.

5.3. Implications for educators

This work also has implications for educators and can help them decide what tool(s) to use in their classroom. Prior work has found that data science instructors favour tools that support diverse online materials, provide explanations and examples, and include classroom activities (Schwab-McCoy, Baker, and Gasper 2021). In including supplemental educational materials as part of our analysis of the tools, we hope to provide a fuller picture for educators of the tools available to them. Furthermore, this analysis can live alongside other efforts to support educators in making informed decisions about tools and curricula they choose for their classrooms, such as the TEC Rubric (Weintrop et al. 2019).

Our aspiration is that this analysis would also assist educators in determining when and where to include data science in the K-12 curriculum. Following the approach seen with K-8 computer science and computational thinking (e.g. Lee, Martin, and Apone 2014), educators are exploring ways to integrate data science across the curriculum rather than create a stand-alone course. In highlighting what tools support importing learner-created data sets or pulling in live datasets that may be relevant to a given science or social studies course, this analysis may help inform when/how data science shows up across the K-12 curriculum.

5.4. Limitations

While this systematic analysis was performed to the best of our ability, it has limitations. The first one is related to locating and extracting the data. We may have missed tools not mentioned in the sources reviewed or published in venues beyond ACM and IEEE. To minimise this limitation, we also used the recommendations of content experts who helped expand the list of tools we explored. The second limitation relates to the difficulty of analyzing a rapidly changing landscape. This analysis only includes tools we could identify at a certain moment. Over time, more tools will be developed. Therefore, as soon as this analysis was complete, it was outdated. This is also the challenge designers and educators face when trying to implement updated and relevant technologies in their curricula.

6. Conclusions

In light of the growing importance of data in society, it is crucial that students understand the role of data in their lives, develop an understanding of what data science is, and have experiences employing foundational data analysis practices. Data analysis and visualisation tools play an increasingly important role in preparing students for the data-driven world around them and are essential in introducing students to the field. The integration of data science tools and block-based environments into K-12 education shows promise in enhancing students' understanding of data science concepts. Therefore, educators, policymakers, and instructional designers should carefully consider the strengths and limitations of these tools when making decisions about their integration into the curriculum.

They must consider that the use of such tools in the curricula may require dedicated training, adjustments of the lesson plans, and learning outcomes. This review advances our understanding and helps us identify gaps and opportunities for creating new and innovative tools. Our hope is that others will also be able to use the review to ensure that all students can benefit from the opportunities these tools offer.

Acknowledgments

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

We would like to thank Jimmy Lin (UMD College of Education) for his contributions as a second coder for block-based programming tools.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research is supported by the National Science Foundation (award #2141655).

References

Barboza, L., R. Mello, M. Modell, and E. S. Teixeira. 2023. "Blockly-DS: Blocks Programming for Data Science with Visual, Statistical, Descriptive and Predictive Analysis." *LAK23: 13th International Learning Analytics and Knowledge Conference*, 644–649. https://doi.org/10.1145/3576050.3576097.

Bargagliotti, A., C. Franklin, P. Arnold, R. Gould, S. Johnson, L. Perez, and D. A. Spangler. 2020. *Pre-K-12 Guidelines for Assessment and Instruction in Statistics Education II (GAISE II)*. Alexandria, VA: American Statistical Association.

Bart, A. C., J. Tibau, D. Kafura, C. A. Shaffer, and E. Tilevich. 2017. "Design and Evaluation of a Block-based Environment with a Data Science Context." *IEEE Transactions on Emerging Topics in Computing* 8 (1): 182–192. https://doi.org/10.1109/TETC.2017.2729585.

Bart, A. C., E. Tilevich, C. A. Shaffer, and D. Kafura. 2015. "Position Paper: From Interest to Usefulness with BlockPy, a Block-based, Educational Environment." *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, 87–89. https://doi.org/10.1109/BLOCKS.2015.7369009.

Batt, S., T. Grealis, O. Harmon, and P. Tomolonis. 2020. "Learning Tableau: A Data Visualization Tool." *The Journal of Economic Education* 51 (3–4): 317–328. https://doi.org/10.1080/00220485.2020.1804503.

Bau, D., J. Gray, C. Kelleher, J. Sheldon, and F. Turbak. 2017. "Learnable Programming: Blocks and Beyond." *Communications of the ACM* 60 (6): 72–80. https://doi.org/10.1145/3015455.

Bisong, E. 2019. "Google Colaboratory." In *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, edited by E. Bisong, 59–64. Apress. https://doi.org/10.1007/978-1-4842-4470-8_7.

Bojer, C. S., and J. P. Meldgaard. 2021. "Kaggle Forecasting Competitions: An Overlooked Learning Opportunity." *International Journal of Forecasting* 37 (2): 587–603. https://doi.org/10.1016/j.ijforecast.2020.07.007.

Brady, C., B. Broll, G. Stein, D. Jean, S. Grover, V. Cateté, T. Barnes, and Á Lédeczi. 2022. "Block-based Abstractions and Expansive Services to Make Advanced Computing Concepts Accessible to Novices." *Journal of Computer Languages* 73:101156. https://doi.org/10.1016/j.cola.2022.101156.

Broll, B., A. Lédeczi, P. Volgyesi, J. Sallai, M. Maroti, A. Carrillo, S. L. Weeden-Wright, C. Vanags, J. D. Swartz, and M. Lu. 2017. "A Visual Programming Environment for Learning Distributed Programming." *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 81–86. https://doi.org/10.1145/3017680.3017741.

The Brown PLT Blog. 2021. *Adding Function Transformers to CODAP*. https://blog.brownplt.org/2021/08/22/codap-transformers.html.

Chandel, S., C. B. Clement, G. Serrato, and N. Sundaresan. 2022. *Training and Evaluating a Jupyter Notebook Data Science Assistant* (arXiv:2201.12901). arXiv. http://arxiv.org/abs/2201.12901.

CODAP. 2022. *CODAP – Common Online Data Analysis Platform*. https://codap.concord.org/.

Dasgupta, S. 2015. "Block-based Programming with Scratch Community Data: A Position Paper." *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, 97–98. https://doi.org/10.1109/BLOCKS.2015.7369011.

Dasgupta, S., and B. M. Hill. 2017. "Scratch Community Blocks: Supporting Children as Data Scientists." *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 3620–3631. https://doi.org/10.1145/3025453.3025847.

DataClassroom. 2022. *DataClassroom*. DataClassroom. https://about.dataclassroom.com.

Deahl, E. 2014. *Better the Data You Know: Developing Youth Data Literacy in Schools and Informal Learning Environments* (SSRN Scholarly Paper 2445621). https://doi.org/10.2139/ssrn.2445621.

Erickson, T. 2016. "Practical Public Online Data: Introducing Tuva and CODAP." *Promoting Understanding of Statistics about Society IASE Roundtable Conference. Promoting Understanding of Statistics about Society*, December 30, https://doi.org/10.52041/SRAP.16601.

Feng, A., E. Tilevich, and W. Feng. 2015. "Block-based Programming Abstractions for Explicit Parallel Computing." *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, 71–75. https://doi.org/10.1109/BLOCKS.2015.7369006.

Fernandez, C., R. De Deus Lopes, and P. Blikstein. 2023. "Programming Representations: Uncovering the Process of Constructing Data Visualizations in a Block-based Programming Environment." *Proceedings of the 2023*

Symposium on Learning, Design and Technology, 11–20. https://doi.org/10.1145/3594781.3594783.

Fitzallen, N. 2020. "Evaluating Data Analysis Software: The Case of Tinkerplots." *Australian Primary Mathematics Classroom* 12 (1): 23–28. https://doi.org/10.3316/informit.134933698874915.

Franklin, C., and A. Bargagliotti. 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.

Fraser, N. 2015. "Ten Things We've Learned from Blockly." *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, 49–50. https://doi.org/10.1109/BLOCKS.2015.7369000.

Frischemeier, D., R. Biehler, S. Podworny, and L. Budde. 2021. "A First Introduction to Data Science Education in Secondary Schools: Teaching and Learning about Data Exploration with CODAP Using Survey Data." *Teaching Statistics* 43 (S1): S182–S189. https://doi.org/10.1111/test.12283.

Gavriilidis, H., F. Henze, E. Tzirita Zacharatou, and V. Markl. 2023. "SheetReader: Efficient Specialized Spreadsheet Parsing." *Information Systems* 115:102183. https://doi.org/10.1016/j.is.2023.102183.

Geise, M. J. 2021. "An Introduction to Tableau as a Data Visualization Tool." *Journal of Computing Sciences in Colleges* 37 (4): 66.

Gould, R. 2021. "Toward Data-scientific Thinking." *Teaching Statistics* 43 (S1). https://doi.org/10.1111/test.12267.

Haq, H. B. U., H. U. R. Kayani, S. K. Toor, S. Zafar, and I. Khalid. 2020. *The Popular Tools of Data Sciences: Benefits, Challenges and Applications*.

Harvey, B., D. D. Garcia, T. Barnes, N. Titterton, O. Miller, D. Armendariz, J. McKinsey, et al. 2014. "Snap! (Build Your own Blocks)." *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 749. https://doi.org/10.1145/2538862.2539022.

Hersh, W. R., R. E. Hoyt, S. Chamberlin, J. S. Ancker, A. Gupta, and T. B. Borlawsky-Payne. 2023. "Beyond Mathematics, Statistics, and Programming: Data Science, Machine Learning, and Artificial Intelligence Competencies and Curricula for Clinicians, Informaticians, Science Journalists, and Researchers." *Health Systems* 12 (3): 255–263. https://doi.org/10.1080/20476965.2023.2237745.

Hoque, M. N., D. Coelho, and K. Mueller. 2019. "Examining the Visualization Practices of Data Scientists on Kaggle." *IEEE VIS*, 20–25. https://naimulh0que.github.io/docs/Kaggle_Analysis.pdf.

IDSSP Curriculum Team. 2019. *Curriculum Frameworks for Introductory Data Science*.

Kelleher, C., J. Maloney, P. Medlock-Walton, E. Patton, and D. Wendel. 2017. "Invited Panel: The Future of Blocks Programming." *2017 IEEE Blocks and Beyond Workshop (B&B)*, 99–101. https://doi.org/10.1109/BLOCKS.2017.8120421.

Khalajzadeh, H., M. Abdelrazek, J. Grundy, J. Hosking, and Q. He. 2022. "Survey and Analysis of Current End-user Data Analytics Tool Support." *IEEE Transactions on Big Data* 8 (1): 152–165. https://doi.org/10.1109/TBDATA.2019.2921774.

Krishnamurthi, S., E. Schanzer, J. G. Politz, B. S. Lerner, K. Fisler, and S. Dooman. 2020. *Data Science as a Route to AI for Middle- and High-school Students* (arXiv:2005.01794). arXiv. http://arxiv.org/abs/2005.01794.

Kross, S., and P. J. Guo. 2019. "Practitioners Teaching Data Science in Industry and Academia: Expectations, Workflows, and Challenges." *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3290605.3300493.

Ladner, R. E., and A. Stefik. 2017. "AccessCSforall: Making Computer Science Accessible to K-12 Students in the United States." *ACM SIGACCESS Accessibility and Computing* 118:3–8. https://doi.org/10.1145/3124144.3124145.

LaMar, T., and J. Boaler. 2021. "The Importance and Emergence of K-12 Data Science." *Phi Delta Kappan* 103 (1): Article 1. https://doi.org/10.1177/00317217211043627.

Le, D.-T. 2013. "Bringing Data to Life Into an Introductory Statistics Course with Gapminder." *Teaching Statistics* 35 (3): 114–122. https://doi.org/10.1111/test.12015.

Lee, V. R., and V. Delaney. 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.

Lee, I., F. Martin, and K. Apone. 2014. "Integrating Computational Thinking Across the K–8 Curriculum." *ACM Inroads* 5 (4): 64–71. https://doi.org/10.1145/2684721.2684736.

Lee, V. R., M. H. Wilkerson, and K. Lanouette. 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.

Lin, Y., and D. Weintrop. 2021. "The Landscape of Block-based Programming: Characteristics of Block-based Environments and How they Support the Transition to Text-based Programming." *Journal of Computer Languages* 67:101075. https://doi.org/10.1016/j.cola.2021.101075.

Lindín, C., K. Steffens, and A. Bartolomé. 2022. "Experiencing Edublocks: A Project to Help Students in Higher Education to Select their Own Learning Paths." *Journal of Interactive Media in Education* 2022 (1): Article 1. https://doi.org/10.5334/jime.731.

Love, J., R. Selker, M. Marsman, T. Jamil, D. Dropmann, J. Verhagen, A. Ly, et al. 2019. "JASP: Graphical Statistical Software for Common Statistical Designs." *Journal of Statistical Software* 88:1–17. https://doi.org/10.18637/jss.v088.i02.

Marriott, K., B. Lee, M. Butler, E. Cutrell, K. Ellis, C. Goncu, M. Hearst, K. McCoy, and D. A. Szafir. 2021. "Inclusive Data Visualization for People with Disabilities: A Call to Action." *Interactions* 28 (3): 47–51. https://doi.org/10.1145/3457875.

Marwan, S., and T. W. Price. 2023. "iSnap: Evolution and Evaluation of a Data-driven Hint System for Block-based Programming." *IEEE Transactions on Learning Technologies* 16 (3): 399–413. https://doi.org/10.1109/TLT.2022.3223577.

McNamara, A. 2019. "Key Attributes of a Modern Statistical Computing Tool." *The American Statistician* 73 (4): 375–384. https://doi.org/10.1080/00031305.2018.1482784.

Meitiner, P., and P. Seneviratne. 2020. "Data Science Goes Digital." In *Beginning Data Science, IoT, and AI on Single Board Computers: Core Skills and Real-world Application with the BBC micro:bit and XinaBox*, edited by P. Meitiner and P. Seneviratne, 23–48. Apress. https://doi.org/10.1007/978-1-4842-5766-1_2.

Microsoft. 2023. *Power BI*. https://powerbi.microsoft.com/en-us/.

Mojica, G. F., C. N. Azmy, and H. S. Lee. 2019. "Exploring Data with CODAP." *The Mathematics Teacher* 112 (6): 473–476. https://doi.org/10.5951/mathteacher.112.6.0473.

Moon, P. F., R. Israel-Fishelson, R. E. Tabak, and D. Weintrop. 2023. "The Tools Being Used to Introduce Youth to Data Science." *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*, 150–159. https://doi.org/10.1145/3585088.3589363.

Nelson, M. J., and A. K. Hoover. 2020. "Notes on Using Google CoLaboratory in AI Education." *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, 533–534. https://doi.org/10.1145/3341525.3393997.

Peng, L. H., M.-H. Bai, and I. Siswanto. 2020. "A Study of Learning Motivation of Senior High Schools by Applying Unity and Mblock on Programming Languages Courses." *Journal of Physics: Conference Series* 1456 (1): 012037. https://doi.org/10.1088/1742-6596/1456/1/012037.

Pimentel, D. R., N. J. Horton, and M. H. Wilkerson. 2022. *Tools to Support Data Analysis and Data Science in k-12 Education*. 22.

Pöhner, N., T. Schmidt, A. Greubel, M. Hennecke, and M. Ehmann. 2019. "BlocklySQL: A New Block-based Editor for SQL." *Proceedings of the 14th Workshop in Primary and Secondary Computing Education*, 1–2. https://doi.org/10.1145/3361721.3362104.

Politz, J. G., K. Fisler, S. Krishnamurthi, and B. S. Lerner. 2018. "From Spreadsheets to Programs: Data Science and CS1 in Pyret." *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* 1058. https://doi.org/10.1145/3159450.3162371.

Poole, M. 2017. "Extending the Design of a Blocks-based Python Environment to Support Complex Types." *2017 IEEE Blocks and Beyond Workshop (B&B)*, 1–7. https://doi.org/10.1109/BLOCKS.2017.8120400.

Price, T. W., and T. Barnes. 2017. "Showpiece: ISnap Demonstration." *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 339–340. https://doi.org/10.1109/VLHCC.2017.8103499.

Price, T. W., Y. Dong, R. Zhi, B. Paaßen, N. Lytle, V. Cateté, and T. Barnes. 2019. "A Comparison of the Quality of Data-driven Programming Hint Generation Algorithms." *International Journal of Artificial Intelligence in Education* 29 (3): 368–395. https://doi.org/10.1007/s40593-019-00177-z.

Reiten, L., and S. Strachota. 2016. "Promoting Statistical Literacy Through Tuva." *The Mathematics Teacher* 110 (3): 228–231. https://doi.org/10.5951/mathteacher.110.3.0228.

Ridgway, J. 2016. "Implications of the Data Revolution for Statistics Education." *International Statistical Review* 84 (3): 528–549. https://doi.org/10.1111/insr.12110.

Rosenberg, J. M., E. H. Schultheis, M. K. Kjelvik, A. Reedy, and O. Sultana. 2022. "Big Data, Big Changes? The Technologies and Sources of Data Used in Science Classrooms." *British Journal of Educational Technology* 53 (5): 1179–1201. https://doi.org/10.1111/bjet.13245.

Saldanha, L., and N. Hatfield. 2021. "Students Conceptualizing the Box Plot as a Tool for Structuring Quantitative Data: A Design Experiment Using TinkerPlots." *Canadian Journal of Science, Mathematics and Technology Education* 21 (4): 758–782. https://doi.org/10.1007/s42330-021-00184-0.

Schanzer, E., N. Pfenning, F. Denny, S. Dooman, J. G. Politz, B. S. Lerner, K. Fisler, and S. Krishnamurthi. 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.

Schwab-McCoy, A., C. M. Baker, and R. E. Gasper. 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.

Sheth, A., S. Padhee, and A. Gyrard. 2019. "Knowledge Graphs and Knowledge Networks: The Story in Brief." *IEEE Internet Computing* 23 (4): 67–75. https://doi.org/10.1109/MIC.2019.2928449.

Silva, Y. N., and J. Chon. 2015. "DBSnap: Learning Database Queries by Snapping Blocks." *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 179–184. https://doi.org/10.1145/2676723.2677220.

Stefik, A., and R. Ladner. 2017. "The Quorum Programming Language." *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 641. https://doi.org/10.1145/3017680.3022377.

Strahler, J., M. Mcquaigue, A. Goncharow, D. Burlinson, K. Subramanian, E. Saule, and J. Payton. 2020. *Real-world Assignments at Scale to Reinforce the Importance of Algorithms and Complexity*. CCSC NE. https://par.nsf.gov/biblio/10158622-real-world-assignments-scale-reinforce-importance-algorithms-complexity.

Tableau: Business intelligence and analytics software. 2023. *Tableau*. https://www.tableau.com/node/62770.

Tock, K. 2019. "Google CoLaboratory as a Platform for Python Coding with Students." *Robotic Telescopes, Student Research and Education (RTSRE) Proceedings* 2 (1): Article 1. https://doi.org/10.32374/rtsre.2019.013.

Wang, L., and S. Dasgupta. 2022. "Dataland: An Informed, Situated, and Critical Approach to Data Literacy." *Proceedings of the 2nd annual meeting of the international society of the learning sciences 2022*.

Watson, J., and J. Donne. 2009. "TinkerPlots as a Research Tool to Explore Student Understanding." *Technology Innovations in Statistics Education* 3 (1). https://doi.org/10.5070/T531000034.

Weintrop, D. 2019. "Block-based Programming in Computer Science Education." *Communications of the ACM* 62 (8): 22–25. https://doi.org/10.1145/3341221

Weintrop, D. 2021. "The Role of Block-based Programming in Computer Science Education." *Understanding Computing Education* 1:71–78.

Weintrop, D., M. Coenraad, J. Palmer, and D. Franklin. 2019. "The Teacher Accessibility, Equity, and Content (TEC)

Rubric for Evaluating Computing Curricula." *ACM Transactions on Computing Education (TOCE)* 20 (1): 1–30.

Weintrop, D., and N. Holbert. 2017. "From Blocks to Text and Back: Programming Patterns in a Dual-modality Environment." *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 633–638. https://doi.org/10.1145/3017680.3017707.

Weintrop, D., D. C. Shepherd, P. Francis, and D. Franklin. 2017. "Blockly Goes to Work: Block-based Programming for Industrial Robots." *2017 IEEE Blocks and Beyond Workshop (B&B)*, 29–36. https://doi.org/10.1109/BLOCKS.2017.8120406.

Weintrop, D., and U. Wilensky. 2018. "How Block-Based, Text-based, and Hybrid Block/Text Modalities Shape Novice Programming Practices." *International Journal of Child-Computer Interaction* 17:83–92. https://doi.org/10.1016/j.ijcci.2018.04.005.

Werner, E., L. Manjunath, J. Frenzel, and S. Torge. 2021. "Bridging between Data Science and Performance Analysis: Tracing of Jupyter Notebooks." *Proceedings of the First International Conference on AI-ML Systems*, 1–7. https://doi.org/10.1145/3486001.3486249.

Wild, C. J. 2018. "Gaining iNZights from Data. Looking Back, Looking Forward." *Proceedings of the 10th International Conference on Teaching Statistics (ICOTS10)*, 1–5. https://iase-web.org/icots/10/proceedings/pdfs/ICOTS10_9A3.pdf.

Wild, C. J., T. Elliott, and A. Sporle. 2021. "On Democratizing Data Science: Some iNZights into Empowering the Many." *Harvard Data Science Review* 3.

Wimmer, H., L. M. Powell, B. Ghosh, C. Snow, D. Hayes, and C. Dwyer. 2016. "A Comparison of Open Source Tools for Data Science." *Journal of Information Systems Applied Research* 9 (2): 4–12.

Appendices

Appendix 1

Table A1. Coding scheme of tools' capabilities (RQ1).

			Data visualisation			Data availability				Extensibility
Tools	Data manipulations	Statistical capabilities	Tabular display	Type of graph	Creation method	Built-in data	Import	API	Export	
Blockly	Aggregating	N/A	×	Line	Code	×	✓	×	✓	✓
Blockly-DS	Filtering, Aggregating	Correlation, Linear Regression, Decision Trees	✓	Scatterplot, Bar Chart, Line Chart, Pie Chart, Box Plot, Histogram	Blocks	×	✓	×	✓	×
BlocklySQL	Filtering, Sorting, Aggregating	N/A	✓	N/A	N/A	✓	×	×	✓	✓
BlockPy	Filtering, Sorting	N/A	×	Scatterplot, Bar Chart, Line Chart, Box Plot, Histogram	Blocks / Code	✓	✓	×	✓	×
Bridges CS	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression	×	Line Chart	Code	✓	✓	×	✓	×
CODAP	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression	✓	Scatterplot, Bar Chart, Histogram, Box Plot, Map	GUI	✓	✓	✓	✓	✓
DataClassroom	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression	✓	Scatterplot, Bar Chart, Line Chart, Pie Chart, Histogram, Box Plot	GUI	✓	✓	×	✓	×
Datacommons	N/A	N/A	✓	Scatterplot, Map	GUI	✓	×	✓	✓	✓
Dataland	Filtering, Sorting, Aggregating	N/A	✓	Scatterplot	Blocks	×	✓	×	✓	×
DBSnap	Filtering, Deleting, Sorting, Aggregating	N/A	✓	Scatterplot, Bar Chart, Line Chart, Pie Chart, Area, Histogram, Bubble Chart	Blocks	✓	✓	×	×	×
EduBlocks	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression	×	Scatterplot, Bar Chart, Stack Chart, Line Chart, Stack Graph, Pie Chart, Radar	Blocks / Code	×	✓	×	×	✓
GapMinder	N/A	N/A	×	Scatterplot	GUI	✓	×	×	✓	×
Google Colab	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression	✓	Line Chart, Bar Chart, Histogram, Scatterplot, Stack Chart, Pie Chart, 3D Graphs	Code	×	✓	✓	✓	×
GP	Filtering, Deleting, Sorting	N/A	×	Scatterplot	Blocks	×	✓	×	✓	×
iNZight	Filtering, Sorting	Correlation, Linear Regression	✓	Bar Chart, Dotplot,						
Box Plot, Grid Density, Hexbin Plot	GUI	✓	✓	✓	✓	×				
iSnap	Filtering, Deleting	N/A	×	Scatterplot, Line Chart	Blocks	×	✓	×	✓	×
JASP	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression, Structural Equation Modeling (SEM), Machine Learning	✓	Scatterplot, Bar, Line Chart, Histogram, Box Plot, Pie Chart	GUI	✓	✓	×	✓	×
Jupyter Notebook	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression	✓	Line, Scatterplot, Bar, Histogram, Box Plot, Violin, Pie Chart, etc.	Code	×	✓	✓	✓	✓

(*Continued*)

Table A1. Continued.

			Data visualisation			Data availability				Extensibility
Tools	Data manipulations	Statistical capabilities	Tabular display	Type of graph	Creation method	Built-in data	Import	API	Export	
kaggle	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression	✓	Scatterplot, Bar, Line Chart, Histogram, Box Plot, Area Plot, Pie Chart	Code	✓	✓	×	✓	×
MakeCode Data Science Editor	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression	✓	Scatterplot, Bar Chart, Line Chart, Histogram, Box Plot, Heatmap	Blocks	✓	✓	×	✓	✓
mBlock	Filtering	N/A	✓	Scatterplot, Bar Chart, Line Chart, Pie Chart	Blocks / Code	✓	✓	×	✓	×
NetsBlox	N/A	N/A	×	Scatterplot, Line Chart, Histogram, Map, Bar Chart	Blocks	×	✓	✓	×	×
PlayData	Filtering, Deleting, Sorting, Aggregating	N/A	✓	Scatterplot	Blocks	×	✓	✓	×	×
Pyret	Filtering, Deleting, Sorting	Correlation, Linear Regression	✓	Scatterplot, Bar Chart, Line Chart, Pie Chart, Histogram	Code	×	✓	×	✓	×
Quorum	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression	×	Scatterplot, Bar Chart, Pie Chart, Line Chart, Histogram, Box Plot, Violin Plots	Code	×	✓	×	✓	✓
Scratch Data Blocks	N/A	N/A	✓	Scatterplot, Line Chart	Blocks	×	✓	×	✓	✓
Snap!	Filtering, Deleting	N/A	✓	Scatterplot, Line Chart	Blocks	×	✓	×	✓	✓
Tableau	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression, Complex Models	✓	Scatterplot, Bar Chart, Line Chart, Pie Chart Histogram, Box Plot	GUI	✓	✓	×	✓	✓
TinkerPlots	Filtering, Deleting, Sorting, Aggregating	Correlation, Linear Regression	✓	Scatterplot, Pie Chart, Histogram	GUI	✓	✓	×	✓	×
Tuva	Filtering, Deleting	Correlation, Linear Regression	✓	Scatterplot, Bar Chart, Line Chart, Pie Chart, Histogram, Box Plot, Map	GUI	✓	✓	×	✓	×

Appendix 2

Table A2. Coding scheme of supported interactions and educational features (RQ2).

Tools	Runtime environment	Programming modality	Scripting language	Instructional materials	Classroom assignments
Blockly	Browser	Read-Only One-Way Transition	Python, Javascript, PHP, Lua, DART	×	×
Blockly-DS	Browser	Block-based	N/A	×	×
BlocklySQL	Browser	Dual Modality	SQL	×	×
BlockPy	Browser	Dual Modality	Python	✓	✓
Bridges CS	Win/Mac	Text-Based	C++, Java, Python	✓	×
CODAP	Browser	GUI	N/A	✓	✓
DataClassroom	Browser	GUI / Text-Based	R	✓	✓
Datacommons	Browser	GUI	N/A	✓	×
Dataland	Browser	Block-based	N/A	×	×
DBSnap	Browser	Read-Only One-Way Transition	SQL	×	×
EduBlocks	Browser		Python	✓	✓

(*Continued*)

Table A2. Continued.

Tools	Runtime environment	Programming modality	Scripting language	Instructional materials	Classroom assignments
		Editable One-Way Transition			
GapMinder	Browser/Win/Mac	GUI	N/A	✓	×
Google Colab	Browser	Text-Based	Python, R	×	×
GP	Browser/Win/Mac	Block-based	N/A	✓	×
iNZight	Browser/Win/Mac/Linux	GUI	N/A	✓	×
iSnap	Browser	Block-based	N/A	×	×
JASP	Browser	GUI	N/A	✓	×
Jupyter Notebook	Browser/Win/Mac/Linux	Text-Based	Python	×	×
kaggle	Browser	Text-Based	R, Python	✓	✓
MakeCode Data Science Editor	Browser	Block-based	N/A	×	×
mBlock	Browser/Win/Mac/iOS/ Android	Read-Only One-Way Transition	Python	✓	✓
NetsBlox	Browser	Block-based	N/A	✓	×
Playdata	Browser	Block-based	N/A	✓	×
Pyret	Browser	Text-Based	Pyret	×	×
Quorum	Browser/Win/Mac	Text-Based	Quorum, MySQL	✓	×
Scratch Data Blocks	Browser	Block-based	N/A	✓	×
Snap!	Browser	Block-based	N/A	✓	×
Tableau	Browser/Win/Mac	GUI	N/A	✓	×
TinkerPlots	Win/Mac	GUI	N/A	✓	×
Tuva	Browser	GUI	N/A	✓	✓

Appendix 3

Table A3. Coding scheme of accessibility features (RQ3).

Tools	Screen reader support	Colour palette	Simplified representations	Customised interactions	Font size modification
Blockly	×	N/A	N/A	×	×
Blockly-DS	×	✓	×	×	×
BlocklySQL	×	N/A	N/A	×	×
BlockPy	×	✓	×	×	×
Bridges CS	×	✓	✓	×	×
CODAP	×	✓	×	×	×
DataClassroom	✓	✓	×	×	✓
Datacommons	×	✓	×	×	×
Dataland	×	✓	×	×	×
DBSnap	×	N/A	N/A	×	×
EduBlocks	×	✓	×	×	✓
GapMinder	×	✓	✓	×	✓
Google Colab	×	✓	×	×	✓
GP	×	N/A	N/A	×	×
iNZight	×	✓	×	×	✓
iSnap	×	N/A	N/A	×	×
JASP	×	✓	×	×	✓
Jupyter Notebook	×	✓	×	×	✓
kaggle	×	✓	×	×	×
MakeCode Data Science Editor	×	✓	×	×	×
mBlock	×	N/A	N/A	×	×
NetsBlox	×	✓	×	×	×
Playdata	×	✓	×	×	×
Pyret	×	✓	×	×	✓
Quorum	×	✓	×	×	×
Scratch Data Blocks	×	N/A	N/A	×	✓
Snap!	×	✓	×	×	×
Tableau	×	✓	×	×	×
TinkerPlots	×	✓	×	×	×
Tuva	✓	✓	×	×	✓

Appendix 4

Table A4. Functions in block-based data science tools (RQ4).

Tools	Ethics	Inquiry with Data	Distributions/ Variability	Measures of Center	Programming – Commenting	Programming – Logic	Programming – Functions	Transformations	Variable Association	Graphs & Figures	Sampling & Simulations	Machine Learning	Time Series Data	Map Data
Blockly	×	×	✓	✓	×	✓	✓	×	×	×	×	×	×	×
Blockly-DS	×	×	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	×	×
BlocklySQL	×	×	×	✓	×	✓	×	✓	×	×	×	×	×	×
BlockPy	×	×	×	×	×	✓	✓	×	×	✓	×	×	×	×
Dataland	×	×	×	×	×	✓	×	✓	×	✓	×	×	×	✓
DBSnap	×	×	×	×	×	✓	×	✓	×	×	×	×	×	×
EduBlocks	×	×	×	×	×	✓	✓	×	×	✓	✓	×	×	×
GP	×	×	×	×	✓	✓	✓	✓	×	×	×	×	×	×
mBlock	×	×	×	×	×	✓	✓	✓	×	×	×	×	×	×
NetsBlox	×	×	×	×	×	✓	✓	✓	×	×	×	×	×	✓
Playdata	×	×	×	✓	×	✓	✓	✓	×	×	×	×	×	×
Scratch Data Blocks	×	×	×	×	×	✓	✓	✓	×	×	×	×	×	×
Snap!	×	×	×	×	×	✓	✓	✓	×	×	×	×	×	×