Journal on Mathematics Education
Volume 16, No. 2, 2025, pp. 423-448

How computational thinking can be integrated in statistical learning: A cuboid framework

Edi Irawan[1], Rizky Rosjanuardi[2,*], Sufyani Prabawanto[3]

[1]Mathematics Education Study Program, Universitas Islam Negeri Kiai Ageng Muhammad Besari, Ponorogo, Indonesia
[2]Mathematics Study Program, Universitas Pendidikan Indonesia, Bandung, Indonesia
[3]Mathematics Education Study Program, Universitas Pendidikan Indonesia, Bandung, Indonesia
*Correspondence: rizky@upi.edu

Received: 3 November 2024 | Revised: 12 February 2025 | Accepted: 11 April 2025 | Published Online: 13 April 2025

Abstract

In the context of an increasingly data-intensive society, the integration of Computational Thinking (CT) into statistics education is essential to prepare students with the analytical and problem-solving competencies required for navigating complex data environments. Despite growing recognition of its importance, existing pedagogical practices frequently lack systematic didactical frameworks that effectively embed CT within statistical learning, particularly in higher education. Addressing this gap, the present study introduces a novel hypothetical didactical design—termed the Cuboid Framework—which systematically integrates CT components into the learning of descriptive statistics using the R programming language in a Google Colab environment. This research employed the Didactical Design Research (DDR) methodology, emphasizing the prospective and metapedadidactic stages to construct and evaluate the framework. Targeted at third-semester undergraduate students enrolled in an introductory statistics course, the Cuboid Framework aligns with learners' developmental levels in both statistical reasoning and CT proficiency. The model is organized as a 5 × 4 × 4 structure, comprising five core statistical tasks, four structured didactical situations (action, formulation, validation, and institutionalization), and four CT elements (decomposition, pattern recognition, abstraction, and algorithmic thinking). Validation procedures included expert review through focus group discussions (FGDs) and an initial classroom implementation followed by metapedadidactic analysis. Findings reveal that the Cuboid Framework fosters a coherent learning progression, enhances students' engagement in statistical inquiry, and supports the development of CT competencies. Classroom observations confirmed that the intentional design of didactical situations facilitates students' cognitive adaptation to computational tasks. While preliminary analyses indicate strong theoretical and practical coherence, further retrospective studies and quantitative evaluations are necessary to ascertain the long-term effects on student learning outcomes. This study contributes a structured and theoretically grounded model for CT integration in statistics education, with implications for improving curriculum design and instructional practice in mathematics education. Future research should aim to test the scalability and efficacy of the Cuboid Framework across diverse educational settings.

Keywords: Computational Thinking, Cuboid Framework, Didactical Design, Statistics Learning, Theory of Didactical Situation

How to Cite: Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2025). How computational thinking can be integrated in statistical learning: A cuboid framework. *Journal on Mathematics Education, 16*(2), 423-448. http://doi.org/10.22342/jme.v16i2.pp423-448

The rapid development of technology in the 21st century has encouraged individuals to master computational thinking (CT) skills as part of the core competencies of this era. CT is an approach to solving complex problems that follows the mindset of computer scientists, involving decomposition, abstraction, pattern recognition, and algorithms (Wing, 2006; 2017). CT is considered an essential skill in the 21st century, alongside reading, writing, and arithmetic (Palts & Pedaste, 2020; Wing, 2017). Due to its broader relevance, CT is important not only for computer science students but also for students across various scientific disciplines (Nouri et al., 2020; So et al., 2020). Therefore, it is crucial to promote and develop students' CT skills, either independently or integrated into existing subjects.

Statistical learning offers significant potential for the development of plugged-in CT skills. Statistics serves as one of the intersections between CT and mathematical thinking (Sneider et al., 2014). Statistics—or what Weintrop et al. (2015) refer to as data practices—are included in the CT taxonomy for science and mathematics. The R programming language, as a tool for programming-based statistical analysis, enables students to deepen their understanding of statistical concepts while simultaneously developing CT skills (Benakli et al., 2017). This demonstrates a strong opportunity to foster students' CT through plugged-in statistics learning using R and relevant statistical tasks.

To date, studies on CT integration have largely been limited to practical implementations, with limited attention to didactic design. Many such studies have not included pedagogical-didactic anticipation in their frameworks. Existing studies generally fall into four main categories. First, several studies have focused on the definition and components of CT (Aho, 2012; Angeli & Giannakos, 2020; Guzdial, 2008; Haseski et al., 2018; Weintrop et al., 2015; Wing, 2006; 2017). Second, a number of studies have analyzed CT trends and research developments using bibliometric analyses (Chen et al., 2023; Ilic et al., 2018; Irawan et al., 2024b; 2024d; Irawan & Herman, 2023; Roig-Vila & Moreno-Isac, 2020; K.-Y. Tang et al., 2020; Tekdal, 2021), scoping reviews (Acevedo-Borrega et al., 2022; Cutumisu et al., 2019), and systematic reviews (Agbo et al., 2019; Irawan et al., 2024a; 2024c; X. Tang et al., 2020). Third, some studies have developed CT learning trajectories (Rich et al., 2018; 2022). Fourth, some research has evaluated students' CT outcomes through plugged and unplugged learning interventions. These include CT assessment (Allsop, 2019; Chen et al., 2017; Prahmana et al., 2024; Purwasih et al., 2024; Zhong et al., 2016), reliability and validity of CT instruments (Korkmaz et al., 2017), effects of augmented reality (Abdul Hanid et al., 2022), and CT development in early childhood (Akiba, 2022). Despite these trends, the specific design of didactic interventions for CT integration, particularly in descriptive statistics, remains underexplored.

Mapping results indicate that research on didactic design—especially for CT integration in statistical learning—is still minimal. However, the integration of CT into statistics learning necessitates thorough planning, including pedagogical-didactic anticipation (Suryadi, 2019). Such anticipation can be enacted through stages of action, formulation, validation, and institutionalization (Brousseau, 2002). Didactic design grounded in the Theory of Didactical Situations (TDS) has been shown to support the achievement of learning objectives and address potential learning barriers (Irawan, 2024). Accordingly, this study raises the question: How can a hypothetical didactic design for integrating CT into descriptive statistics be constructed based on the TDS framework?

In contrast to earlier studies that focus predominantly on theoretical discourse or application-based integration of CT, this research presents a systematically constructed didactic framework grounded in the TDS. By emphasizing pedagogical-didactic anticipation, this framework integrates CT into descriptive statistics instruction while also anticipating potential learning challenges to improve students' conceptual understanding. Furthermore, this study offers a replicable model adaptable to diverse educational contexts—including schools with limited computational infrastructure—by utilizing cloud-based tools such as Google Colab.

This study aims to bridge the gap in CT integration by proposing a didactic framework that supports a structured, problem-based approach to learning descriptive statistics using R. A well-formulated didactic design is key to maximizing the use of R in Google Colab, allowing students to engage in statistical exploration while developing CT skills such as abstraction, decomposition, and algorithmic reasoning. Through this integrated approach, CT development in statistical learning is expected to reinforce students' understanding of statistical concepts while nurturing essential 21st-century problem-solving competencies.

Cuboid Framework

A hypothetical didactical design was developed to ensure that the processes of knowledge dissemination and acquisition occurred effectively within the learning environment. This design was constructed based on three foundational pillars: the TDS, the four core components of CT, and a structured progression of descriptive statistics learning tasks.

First, TDS, as proposed by Brousseau (2002), delineates four principal learning situations: action, formulation, validation, and institutionalization. These stages guide the pedagogical flow and anticipation of students' learning responses. Second, CT encompasses four key components: decomposition, pattern recognition, abstraction, and algorithmic thinking (Dong et al., 2019; Wing, 2017), each of which plays a critical role in developing problem-solving and analytical skills. Third, the domain of descriptive statistics is divided into five sequential learning tasks: (1) an introduction to descriptive statistics, (2) data presentation, (3) measures of central tendency, (4) measures of data dispersion, and (5) measures of data position (Raykov & Marcoulides, 2013). Integrating these three dimensions, a hypothetical didactical design model referred to as the Cuboid Framework was developed. This model adopts a three-dimensional structure with dimensions of 5 × 4 × 4, as illustrated in Figure 1.

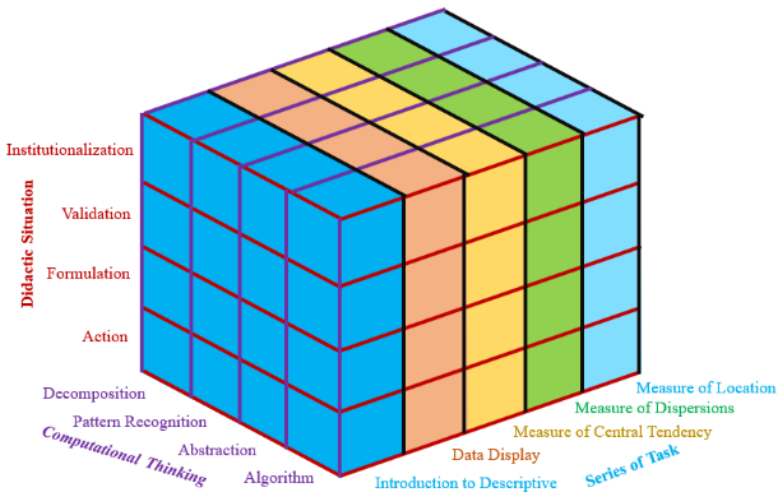


Figure 1. Cuboid framework for didactical design integrating CT into descriptive statistics learning with R

The Cuboid Framework, as shown in Figure 1, is a conceptual three-dimensional model that maps the integration of CT components with didactical stages and statistical content in an R-based learning environment. It is specifically designed to support students' CT development within the context of descriptive statistics learning. The five layers of the framework represent the sequential learning tasks; each linked with corresponding didactical and CT components. The first layer outlines the didactical design for Task 1, which introduces fundamental statistical concepts such as data types, scales, and variables. The second layer corresponds to Task 2, which focuses on data presentation using R in Google Colab. The third layer covers Task 3, where students compute measures of central tendency (mean, median, and mode) using R. The fourth layer represents Task 4, guiding students in calculating measures of data dispersion, including range, mean deviation, variance, and standard deviation. The fifth layer addresses Task 5, where students analyze measures of data position such as percentiles and quartiles.

The Cuboid Framework offers a systematic and pedagogically informed approach to facilitating CT within statistical learning. It provides structured guidance that aligns instructional content with CT components across different didactical situations, leveraging the capabilities of R programming through a cloud-based platform (Google Colab). This model not only fosters a deeper understanding of statistical concepts but also enhances students' computational thinking in an integrated, contextually rich learning environment.

METHODS

This study focuses on the development of a hypothetical didactical design that integrates CT into descriptive statistics instruction using the R programming language. The design aims to support effective and meaningful learning tailored to the needs of third-semester undergraduate students enrolled in the Basic Statistics course within the Mathematics Education program at IAIN Ponorogo. This didactical design was formulated as part of the instructional planning process, emphasizing three interrelated components essential to the learning process: (1) the components of CT, (2) the theoretical foundation of the didactical situation, and (3) the learning objectives of descriptive statistics. Furthermore, the CT components incorporated in the design include decomposition, abstraction, pattern recognition, and algorithmic thinking (Wing, 2017; Dong et al., 2019). These elements are essential for enhancing students' problem-solving and reasoning skills in statistical contexts. The didactical structure follows the TDS, which comprises four learning situations: action, formulation, validation, and institutionalization (Brousseau, 2002). The content domain of descriptive statistics includes foundational competencies such as understanding data and variables, data collection methods, data presentation techniques, measures of central tendency, measures of dispersion, and measures of data position (Raykov & Marcoulides, 2013).

A qualitative research approach was employed, adopting the Didactical Design Research (DDR) framework proposed by Suryadi (2019). DDR is particularly well-suited for achieving the objectives of this study, which center on the development of a hypothetical didactical design that integrates CT into descriptive statistics learning using the R software environment in Google Colab. The adoption of DDR also aligns with the critical-interpretive paradigm underpinning the research. The DDR framework consists of three interrelated phases: prospective analysis, metapedadidactic analysis, and retrospective analysis (Suryadi, 2013). In the prospective analysis, the initial stage, the researcher designed both a hypothetical learning trajectory (HLT) and a hypothetical didactical design. The HLT outlines the expected progression of student learning and anticipates how CT elements and statistical concepts will be introduced and developed. The metapedadidactic analysis involved the classroom implementation of the previously constructed HLT and hypothetical design, with a focus on observing how students utilized CT in analyzing descriptive statistics using R. The retrospective analysis, which focuses on interpreting students' learning outcomes and the development of CT competencies, was not the primary focus of this study. This research is thus limited to the prospective and metapedadidactic phases, concentrating on the design and implementation of the hypothetical didactical framework.

To ensure the credibility and validity of the design, a focus group discussion (FGD) was conducted involving five experts: two mathematics education specialists, two didactical design experts, and one statistics education expert. All participants had substantial academic and research experience in their respective domains. The FGD was structured into three phases: Initial Presentation – Introduction of the preliminary Cuboid Framework and hypothetical didactical design, Critical Discussion – Expert feedback and critique of the conceptual and technical dimensions of the design, and Refinement – Revision of the design based on expert recommendations. To guide the expert evaluation process, the following validation framework was used (see Table 1).

Table 1. Aspect and questions for validations of the hypothetical didactical design

No	Aspect	Questions
1	Alignment with Research Objectives	How well does the didactical design align with the research objectives?
2	CT Integration	Does the design effectively integrate the components of CT?
3	Theoretical and Pedagogical Coherence	To what extent does the design follow HLT and TDS?
4	Feasibility of Classroom Implementation	Can the design be implemented feasibly in actual classroom settings?
5	Clarity, Coherence, and Practicality	How can the design be refined to enhance clarity, coherence, and practicality?

As this study involved expert consultation and did not collect personal or sensitive data, formal ethical clearance was not required. Informed consent was obtained from all participants after providing them with a clear explanation of the study's objectives and procedures. All responses were anonymized to ensure confidentiality, and no personally identifiable information was collected.

RESULTS AND DISCUSSION

In accordance with the stages of DDR, the first phase of this study was the prospective analysis. During this phase, after the initial cuboid framework was developed, it underwent validation by experts through an FGD. All experts agreed that the framework fulfilled the critical aspects outlined in Table 1. However, they also provided several constructive suggestions to improve the didactical design, as summarized in Table 2.

Table 2. Comments and suggestions from the experts

No	Suggestions	Follow-Up Actions
1	Emphasize the use of R in all didactical situations	Integrated the use of R software into the action, formulation, validation, and institutionalization phases.
2	Include statistical context in each didactical situation	Contextualized each phase with relevant statistical content.
3	Improve visualization of the cuboid framework	Revised visual design of the framework to enhance clarity and attractiveness.

These expert recommendations were used to revise and refine the hypothetical didactical design. Once finalized, the design was implemented across five classroom meetings, representing the metapedadidactic analysis stage of the DDR process. An example of the classroom implementation is depicted in Figure 2.

Figure 2. Implementation of the didactical design integrating CT in descriptive statistics learning using R

Figure 2 illustrates the classroom application of the didactical design in a computer laboratory setting, utilizing the R programming language. The instructional implementation was divided into five tasks. Below, we present the detailed didactical design and evidence from its implementation for Task 1: Introduction to Descriptive Statistics.

Didactical Design of Task 1: Introduction to Descriptive Statistics

Task 1 aimed to introduce students to fundamental concepts in descriptive statistics by integrating components of CT with R programming. The design employed a 4×4 framework that combines four CT components—Decomposition, Pattern Recognition, Abstraction, and Algorithm Design—with four phases of Didactical Situations based on TDS: Action, Formulation, Validation, and Institutionalization, as shown in Figure 3.

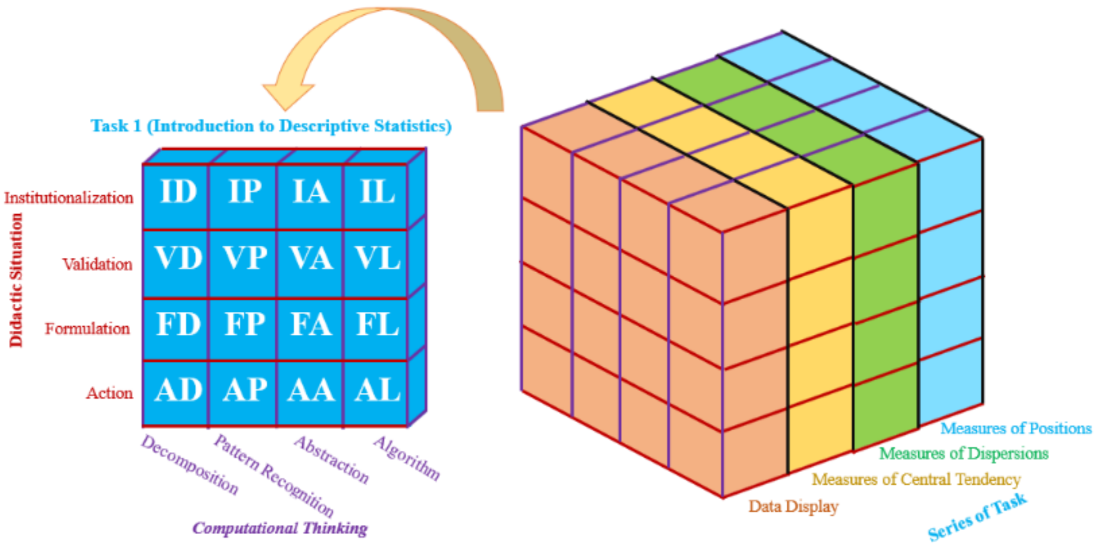


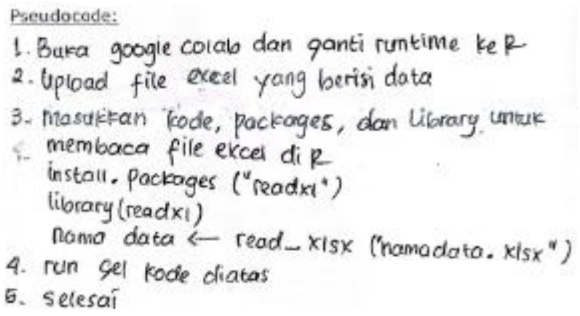
Figure 3. Framework of didactical situations for task 1 (Introduction to descriptive statistics)

Figure 3 is a framework of didactical situations in Task 1, which is introductory descriptive statistics material. The detailed design of activities in each didactical situation is presented in Table 3.

Table 3. Didactical situation in task 1 (Introduction to descriptive statistics)

CT Component	Action Situation	Formulation Situation	Validation Situation	Institutionalization Situation
Decomposition	Complex problems related to data, types of data, scale, and R are decomposed into simpler problems.	Formulate strategies for decomposing complex problems related to data, types of data, scale, and R into simpler problems.	Test the correctness of the strategies for decomposing complex problems related to data, types of data, scale, and R into simpler problems.	Decompose complex problems related to the data, data types, scale, and R into simpler problems with other relevant problems.
Pattern Recognition	Recognize problem-solving patterns related to data, data types, scales, and R.	Develop problem-solving patterns related to data, data types, scales, and R.	Test the correctness of problem-solving patterns related to data, data types, scales, and R.	Recognize problem-solving patterns related to data, data types, scale, and R for other relevant problems.
Abstraction	Simplify the data by selecting relevant information and ignoring irrelevant information related to data, data types, scales, and R.	Formulate data simplification strategies by selecting relevant information and ignoring irrelevant information related to the data, data types, scales, and R.	The correctness of the data simplification strategies is tested by selecting relevant information and ignoring irrelevant information related to data, data types, scales, and R.	Simplify the data by selecting relevant information and ignoring irrelevant information related to data, data type, scale, and R on other relevant issues.
Algorithm	Identify various stages of problem solving related to data, data types, scales, and R.	Develop steps to solve problems related to data, data types, scales, and R.	Test the correctness and effectiveness of using the steps to solve problems related to data, data types, scales, and R.	Use correct and effective steps to solve problems related to data, data types, scales, and R for other relevant problems.

The implementation began with the action situation, where students were tasked with entering data into R using Google Colab. In the formulation situation, they designed a pseudocode outlining the steps to import data into R, as shown in Figure 4.



Translation
1. Open Google Colab and change runtime to R
2. Upload an Excel file containing data
3. Enter code, packages, and library to read Excel files in R
   Install.packages("readxl")
   library(readxl)
   Data name <- read_xlsx ("namadata.xlsx")
4. Run the code cell above
5. End

Figure 4. Example of student pseudocode on task 1 (Entering data in R)

Following this code presented in Figure 4, in the validation situation, students executed the pseudocode in Google Colab to test its correctness. Finally, in the institutionalization situation, they were presented with a new and more complex dataset and asked to apply their knowledge to import it using R. An example of student-generated artifacts from this process is displayed in Figure 5.

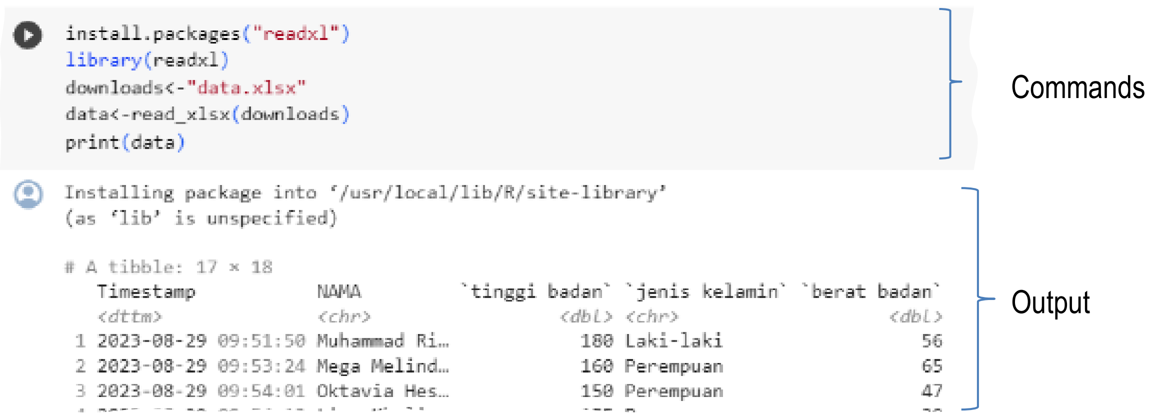


Figure 5. Example of student artifact in task 1 (Entering data in R using Google Colab)

Figure 5 illustrates the process of entering data in R in Google Colab in five steps. First, the "readxl" package was installed. Second, the “readxl” library was activated. Third, Excel data were entered into Google Colab. Fourth, read the Excel data. Finally, the data were displayed using the data using the "print" function. As a result, the data from Excel were successfully entered into R in Google Colab.

Didactical Design in Task 2 (Data Display)

The didactical design in Task 2 integrates Computational Thinking (CT) into the learning of data presentation. Like Task 1, this design consists of 16 didactical situations (4 × 4), structured around the four core CT components—Decomposition, Pattern Recognition, Abstraction, and Algorithm—distributed across the four didactical phases: Action, Formulation, Validation, and Institutionalization. The framework of the didactical situations for Task 2 is depicted in Figure 6.

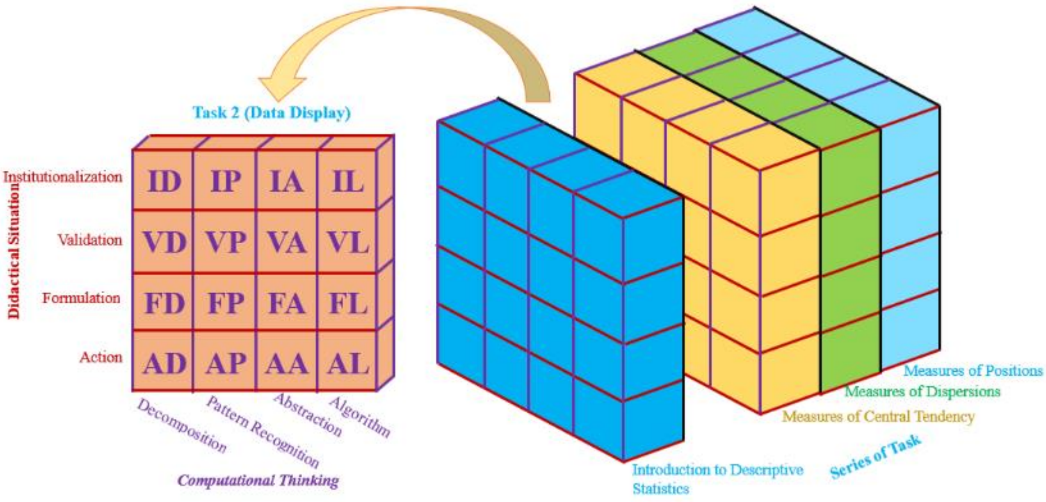


Figure 6. Framework of the didactical situation for task 2 (Data display)

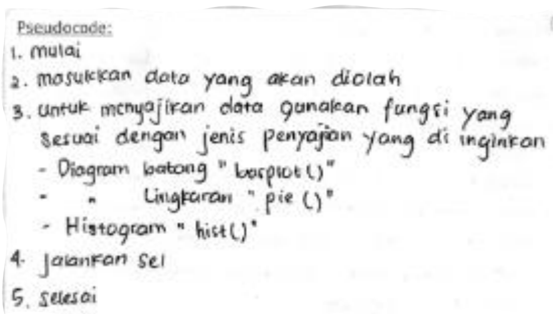
Figure 6 illustrates the conceptual design of the CT-integrated didactical situations for data presentation. The detailed learning activities associated with each didactical situation are outlined in Table 4.

Table 4. Didactical situation in task 2 (Data display)

CT Component	Action Situation	Formulation Situation	Validation Situation	Institutionalization Situation
Decomposition	Decompose complex problems related to data presentation using R into simpler problems.	Formulate a strategy to decompose complex problems related to data presentation using R into simpler problems.	Test the correctness of the strategy of decomposing complex problems related to the presentation of data using R into simpler problems.	Decompose complex problems related to data presentation using R into simpler problems on other relevant problems.
Pattern Recognition	Recognize problem-solving patterns related to data presentation using R.	Develop problem solving patterns related to data presentation using R.	Test the correctness of problem-solving patterns related to data presentation using R.	Recognize problem solving patterns related to the presentation of data using R in other relevant problems.
Abstraction	Simplify data by selecting relevant information and	Formulate data simplification strategies by selecting	Testing the correctness of data simplification strategies by	Simplify data by selecting relevant information and ignoring irrelevant

CT Component	Action Situation	Formulation Situation	Validation Situation	Institutionaliza-tion Situation
	ignoring irrelevant information related to data presentation using R.	relevant information and ignoring irrelevant information related to data presentation using R.	selecting relevant information and ignoring irrelevant information related to data presentation using R.	information related to the presentation of data using R on other relevant problems.
Algorithm	Identify various stages of problem solving related to data presentation using R.	Develop problem solving steps related to data presentation using R.	Testing the correctness and effectiveness of using steps to solve problems related to data presentation using R.	Use correct and effective steps to solve problems related to data presentation using R for other relevant problems.

The implementation of the didactical design in Task 2 commenced with the action situation, wherein students were tasked with presenting a dataset using R in the Google Colab environment. In the formulation situation, students were guided to construct a structured pseudocode for data visualization, exemplified in Figure 7.


Pseudocode:
1. mulai
2. masukkan data yang akan diolah
3. untuk menyajikan data gunakan fungsi yang sesuai dengan jenis penyajian yang di inginkan
- Diagram batang " barplot()"
- " Lingkaran " pie()"
- Histogram " hist()"
4. jalankan sel
5. selesai

Translation
1. Start
2. Enter the data to be processed
3. To present data use the appropriate function with the desired type of data presentation
    - Bar chart "barplot()"
    - Pie chart "pie()"
    - Histogram "hist()"
4. Run Cell
5. End

Figure 7. Example of data presentation pseudocode using R

After successfully developing the data display stages in R, as presented in Figure 7, in the validation situation, students tried to test the pseudocode directly in R. Finally, in the institutionalization situation, students were challenged to solve a new problem, namely, a new, more complex data display in R on Google Colab. An example of the results of the data display artifacts in R on Google Colab produced by the students is presented in Figure 8.

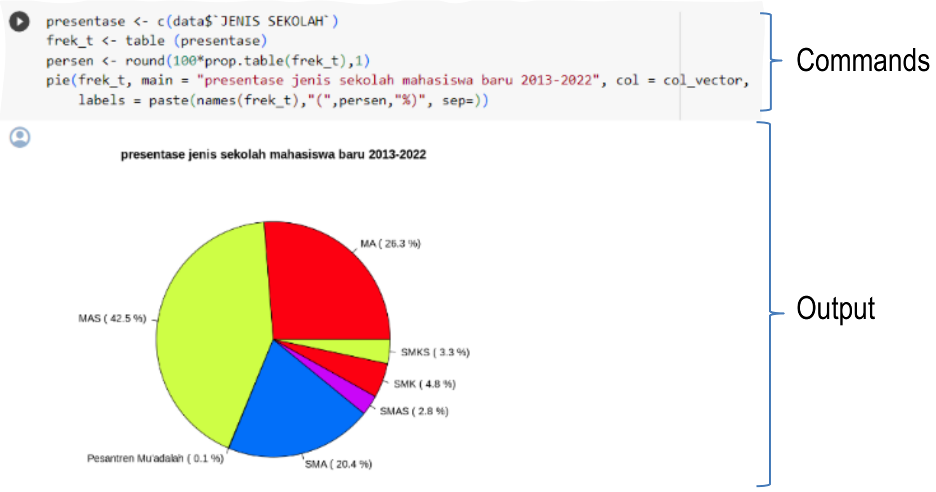


Figure 8. Examples of data visualization artifacts using R

Figure 8 showcases an example of a student-generated artifact, demonstrating the process of data presentation using a pie chart in R within Google Colab in four steps. First, the data were read from a data frame. Second, the frequency of each category was calculated using the "table" function. Third, calculate the percentage using the "prop.table" function and round the results using the "round" function. Finally, the data were displayed in the form of a pie chart using the "pie" function. As a result, R successfully rendered a pie chart representing the categorical distribution of the dataset. This activity illustrates students' ability to apply CT principles in organizing and executing data visualization tasks within a statistical programming context.

Didactical Design in Task 3 (Measures of Central Tendency)

The didactical design at Task 3 is a CT integration didactical design for determining the measure of data concentration, which includes the mean, median, and mode. Figure 9 shows the 4 × 4 framework of the didactical situation, with an emphasis on the CT component adapted to the concept of central tendency.

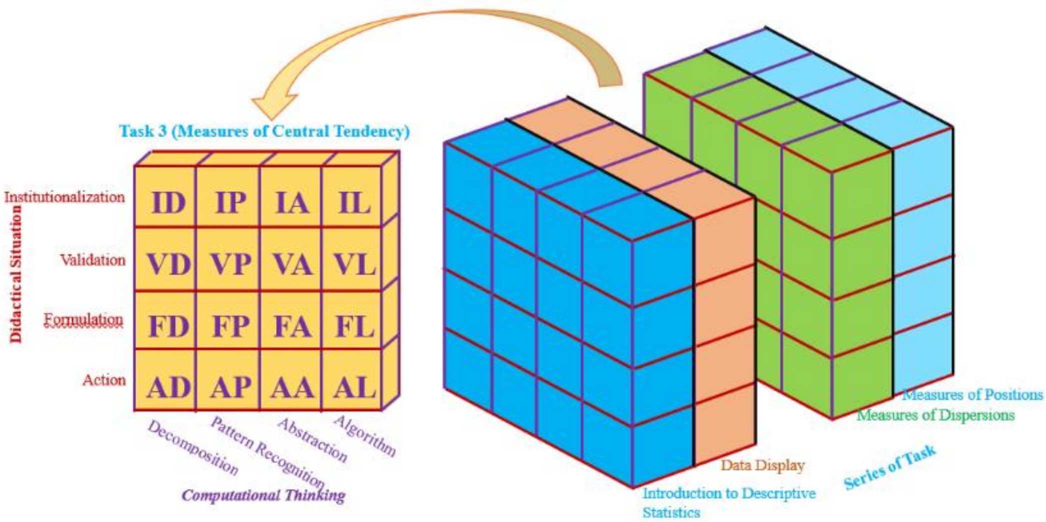


Figure 9. Framework of didactical situations on task 3 (Measures of central tendency)

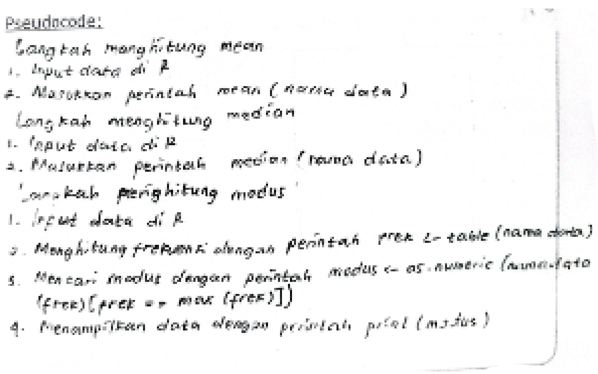
Figure 9 is a framework of didactical situations on Task 3, namely on the material of measures of central tendency, which includes mean, median, and mode. The detailed activity design for each didactical situation is presented in Table 5.

Table 5. Didactical situation on task 3 (Measures of central tendency)

CT Component	Action Situation	Formulation Situation	Validation Situation	Institutionaliza-tion Situation
Decomposition	Decompose complex problems related to finding measures of central tendency (mean, median, and mode) using R into simpler problems.	Formulate a strategy to decompose a complex problem related to finding the measures of central tendency (mean, median, and mode) using R into simpler problems.	Test the correctness of the strategy for decomposing a complex problem related to finding the size measures of central tendency (mean, median, and mode) using R into a simpler problem.	Decompose a complex problem related to finding the measures of central tendency (mean, median, and mode) using R into simpler problems on other relevant problems.
Pattern Recognition	Recognize problem-solving patterns related to measures of central tendency (mean, median, and mode) using R.	Develop problem solving patterns related to finding the size of measures of central tendency (mean, median, and mode) using R.	Test the correctness of the problem-solving pattern related to finding the size of measures of central tendency (mean, median, and mode) using R.	Recognize the problem-solving pattern related to finding the size of measures of central tendency (mean, median, and mode) using R in other relevant problems.
Abstraction	Simplify data by selecting relevant information and ignoring irrelevant information related to finding measures of central tendency (mean, median, and mode) using R.	Formulate data simplification strategies by selecting relevant information and ignoring irrelevant information related to finding measures of central tendency (mean, median, and mode) using R.	Test the correctness of data simplification strategies by selecting relevant information and ignoring irrelevant information related to finding measures of central tendency (mean, median, and mode) using R.	Simplify data by selecting relevant information and ignoring irrelevant information related to finding measures of central tendency (mean, median, and mode) using R on other

CT Component	Action Situation	Formulation Situation	Validation Situation	Institutionalization Situation
				relevant problems.
Algorithm	Identify various stages of problem solving related to finding measures of central tendency (mean, median, and mode) using R.	Develop problem solving steps related to finding measures of central tendency (mean, median, and mode) using R.	Testing the correctness and effectiveness of the use of problem-solving steps related to finding measures of central tendency (mean, median, and mode) using R.	Use correct and effective steps to solve problems related to finding the measures of central tendency (mean, median, and mode) using R on other relevant problems.

The implementation of didactical design in Task 3 starts from the action situation where students are faced with the task of finding data-centering measures that include the mean, median, and mode in R on Google Colab. Then, in the formulation situation, students try to arrange the flow to find the measure of data concentration in R, as reflected in the pseudocode presented in Figure 10.


Pseudocode:
Langkah menghitung mean
1. Input data di R
2. Masukkan perintah mean (nama data)
Langkah menghitung median
1. Input data di R
2. Masukkan perintah median (nama data)
Langkah menghitung modus
1. Input data di R
2. Menghitung frekuensi dengan perintah frek <- table (nama data)
3. Mencari modus dengan perintah modus <- as.numeric (nama data (frek)[frek == max (frek)]])
4. Menampilkan data dengan perintah print (modus)

Translation
Steps to calculate the mean
1. Data input in R
2. Enter the command mean (data name)

Steps to calculate the median
1. Data input in R
2. Enter the median command (data name)

Steps to calculate the mode
1. Data input in R
2. Calculate frequencies with <-table frek command (data name)
3. Searching for a mode with the mode command <- as.numeric(dataname)(frek)(frek = = max(frek)])
4. Displaying data with the print command (mode)

Figure 10. Pseudocode example of calculating mean, median, and mode using R

After successfully compiling the steps to calculate the size of data concentration in R, as presented in Figure 10, in the validation situation, students attempted to test the pseudocode directly in R. Finally, in the institutionalization situation, students were challenged to solve a new problem, namely, finding more complex data concentration measures in R on Google Colab. An example of the artifact of finding the size of the data concentration in R on Google Colab produced by the students is presented in Figure 11.

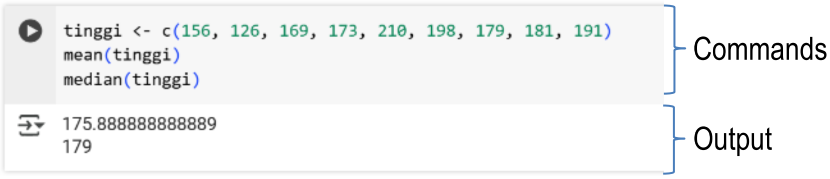


Figure 11. Artifact example of calculating mean and median using R

Figure 11 illustrates the process of calculating the mean and median using R on Google Colab in two steps. First, the data were entered directly. Second, the mean was calculated using the "mean" command and finding the median using the "median" command. As a result, R will present the mean and median of the previously entered data.

Didactical Design in Task 4 (Measures of Dispersion)

The didactical design at Task 4 was a CT integration didactical design used to determine the measures of dispersion, which included range, variance, and standard deviation. There are 16 (4 × 4) CT integration didactical situations on the measures of the dispersion material as the didactical situation framework listed in Figure 12.

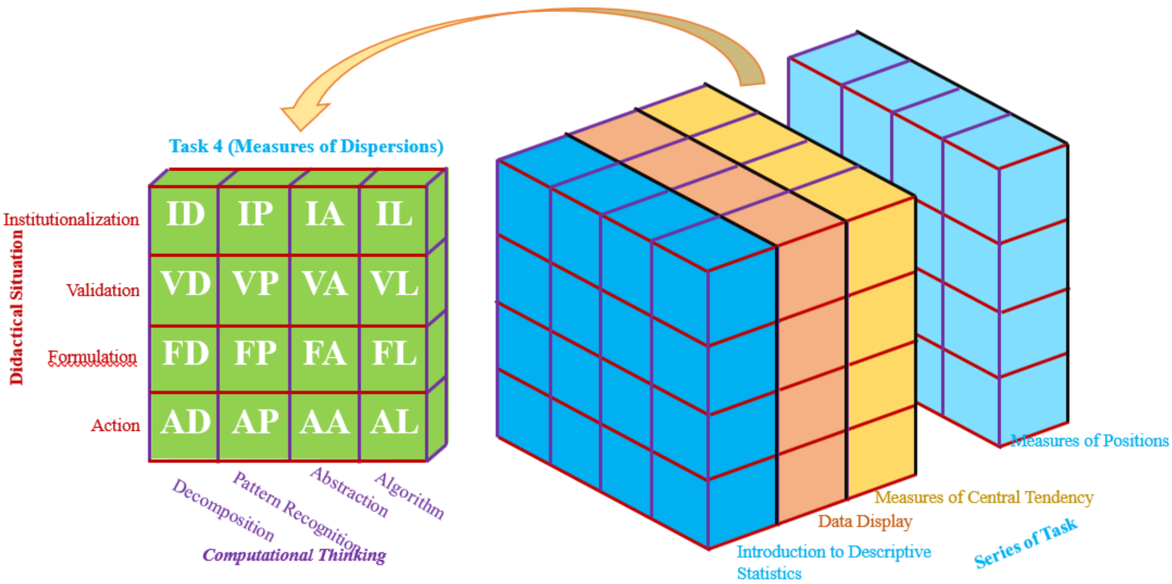


Figure 12. Framework of didactical situations on task 4 (Measures of dispersion)

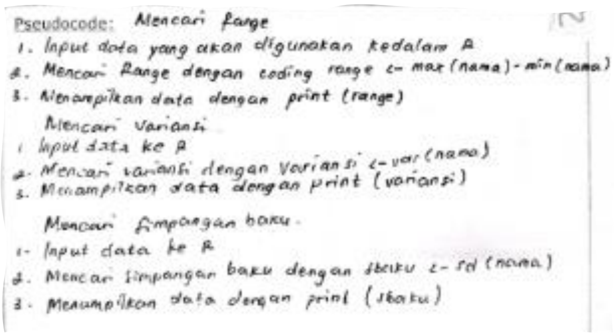
Figure 12 is a framework of didactical situations on Task 4, namely on the material of measures of dispersion, which includes range, variance, and standard deviation. The activity design for each didactical situation is presented in Table 6.

Table 6. Didactical situation in task 4 (Measures of dispersion)

CT Component	Action Situation	Formulation Situation	Validation Situation	Institutionalization Situation
Decomposition	Decompose complex problems related	Formulate a strategy to decompose a	Test the correctness of the strategy of	Decompose a complex problem related to finding

CT Component	Action Situation	Formulation Situation	Validation Situation	Institutionalization Situation
	to finding the size of the spread using R into simpler problems.	complex problem related to finding the size of the spread using R into simpler problems.	decomposing a complex problem related to finding the size of the spread using R into simpler problems.	the size of the spread using R into simpler problems on other relevant problems.
Pattern Recognition	Recognize problem solving patterns related to finding the size of the spread using R.	Develop problem solving patterns related to finding the size of the spread using R.	Test the correctness of problem-solving patterns related to finding the size of the spread using R.	Recognize problem solving patterns related to finding the size of the spread using R on other relevant problems.
Abstraction	Simplifying data by selecting relevant information and ignoring irrelevant information related to finding the size of the spread using R.	Formulate a data simplification strategy by selecting relevant information and ignoring irrelevant information related to finding the size of the spread using R.	Testing the correctness of the data simplification strategy by selecting relevant information and ignoring irrelevant information related to finding the size of the spread using R.	Simplify data by selecting relevant information and ignoring irrelevant information related to finding the size of the spread using R on other relevant problems.
Algorithm	Identify various stages of problem solving related to finding the size of the spread using R.	Develop problem solving steps related to finding the size of the spread using R.	Testing the correctness and effectiveness of the use of problem-solving steps related to finding the size of the spread using R.	Use correct and effective steps to solve problems related to finding the size of the spread using R for other relevant problems.

The implementation of didactical design in Task 4 starts from the action situation where students are faced with the task of calculating the size of data distribution, which includes range, variance, and standard deviation in R on Google Colab. Then, in the formulation situation, students try to arrange the flow to calculate the size of the data distribution in R, as reflected in the pseudocode presented in Figure 13.


Pseudocode: Mencari Range
1. Input data yang akan digunakan kedalam R
2. Mencari Range dengan coding range <- max(nama)-min(nama)
3. Menampilkan data dengan print (range)
Mencari Variansi
1. Input data ke R
2. Mencari variansi dengan Variansi <- var(nama)
3. Menampilkan data dengan print (variansi)
Mencari Simpangan baku.
1. Input data ke R
2. Mencari Simpangan baku dengan sbaku <- sd (nama)
3. Menampilkan data dengan print (sbaku)

Translation
Find a range
1. Input data to be used in R
2. Find a range with coding range <- max(name) – min(name)
3. Displaying data with print(range)

Search for variance
1. Input data to R
2. Find a variance with variance <- var(name)
3. Displaying data with print(variance)

Finding standard deviation
1. Input data to R
2. Find standard deviation with default <-sd(name)
3. Displaying data with print(default)

Figure 13. Example of pseudocode calculating range, variance, and standard deviation

After successfully compiling the steps to calculate the data distribution measure in R, as presented in Figure 13, in the validation situation, students tried to test the pseudocode directly in R. Finally, in the institutionalization situation, students were challenged to solve a new problem, namely, calculating a more complex data distribution measure in R on Google Colab. An example of the artifact of finding the size of the data concentration in R on Google Colab produced by the students is presented in Figure 14.

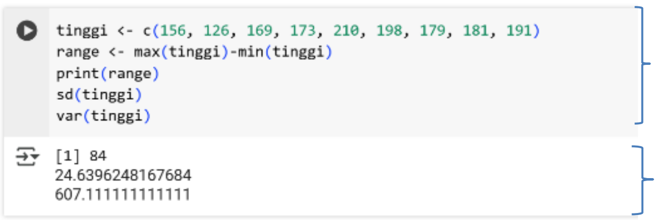

```
tinggi <- c(156, 126, 169, 173, 210, 198, 179, 181, 191)
range <- max(tinggi)-min(tinggi)
print(range)
sd(tinggi)
var(tinggi)
```
```
[1] 84
24.6396248167684
607.111111111111
```


Figure 14. Artifact example of calculating range, variance, and standard using R

Figure 14 illustrates the process of finding the range, variance, and standard deviation using R in Google Colab in five steps. First, the data were entered into R. Second, the range was determined by determining the difference between the highest data using the "max" function and the lowest data using the "min" function. Third, the range was displayed using the "print" function. Fourth, the standard deviation was determined using the "sd" function. Finally, the variance was found using the "var" function. As a result, R presents the range, variance, and standard deviation of the previously entered data.

Didactical Design in Task 5 (Measures of Location)

The didactical design in Task 5 was a CT integration didactical design for determining the size of the data location, which included quartiles, deciles, and percentiles. There are a total of 16 (4 × 4) CT integration didactical situations for the data location size material, as presented in Figure 15.

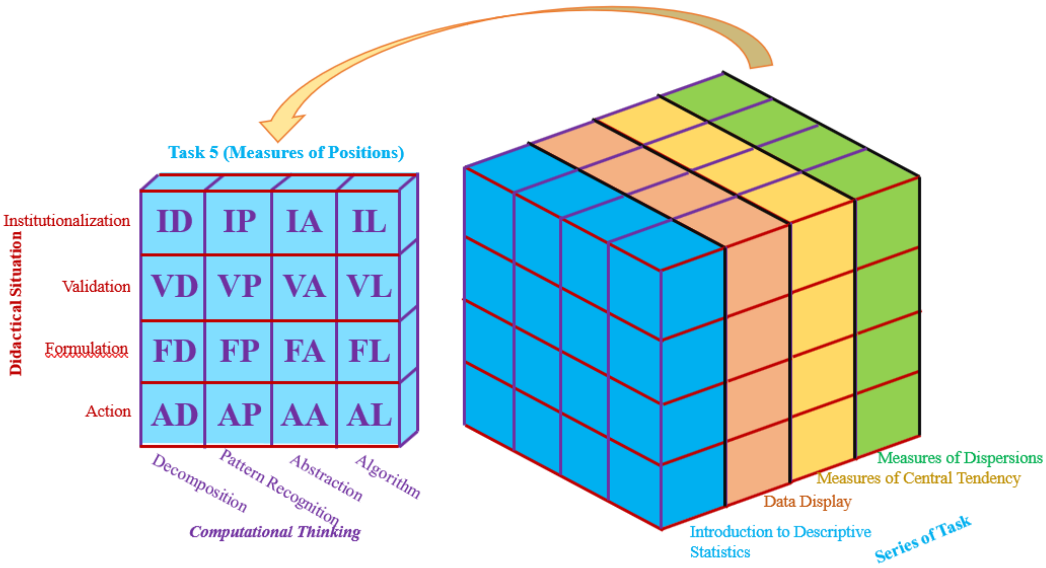


Figure 15. Framework of didactical situation on task 5 (Measures of location)

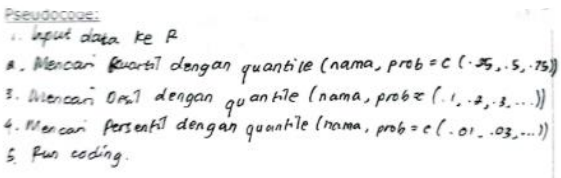
Figure 15 shows the framework of the didactical situation on Task 5, which is on the measures of data location, which includes quartiles, deciles, and percentiles. The detailed activity design for each didactical situation is presented in Table 7.

Table 7. Didactical situation in task 5 (Measures of location)

CT Component	Action Situation	Formulation Situation	Validation Situation	Institutionaliza-tion Situation
Decomposition	Decompose complex problems related to locating using R into simpler problems.	Formulate a strategy to decompose a complex problem related to finding the size of the location using R into simpler problems.	Test the correctness of the strategy to decompose a complex problem related to finding the size of the location using R into simpler problems.	Decompose a complex problem related to locating using R into simpler problems on other relevant problems.
Pattern Recognition	Recognize problem-solving patterns related to locality search using R.	Develop problem solving patterns related to finding the size of the location using R.	Test the correctness of problem-solving patterns related to finding the size of the location using R.	Recognize problem solving patterns related to finding the size of the location using R in other relevant problems.
Abstraction	Simplify data by selecting relevant information and	Formulate data simplification strategies by selecting relevant	Testing the correctness of the data simplification strategy by	Simplify data by selecting relevant information and ignoring irrelevant

CT Component	Action Situation	Formulation Situation	Validation Situation	Institutionaliza-tion Situation
	ignoring irrelevant information related to locality size search using R.	information and ignoring irrelevant information related to finding the size of the location using R.	selecting relevant information and ignoring irrelevant information related to finding the size of the location using R.	information related to finding the size of the location using R on other relevant problems.
Algorithm	Identify various stages of problem solving related to finding the size of the location using R.	Develop problem solving steps related to finding the size of the location using R.	Testing the correctness and effectiveness of the use of problem-solving steps related to finding the size of the location using R.	Use correct and effective steps to solve problems related to finding the size of the location using R for other relevant problems.

The implementation of didactical design in Task 5 starts from the action situation where students were faced with the task of calculating the size of data location, which includes quartiles, deciles, and percentiles in R on Google Colab. Then, in the formulation situation, the students tried to arrange the flow to calculate the size of the data location in R, as reflected in the pseudocode presented in Figure 16.


Pseudocode:
1. Input data ke R
2. Mencari Kuartil dengan quantile (nama, prob = c (.25, .5, .75))
3. Mencari Desil dengan quantile (nama, prob = (.1, .2, .3, ...))
4. Mencari Persentil dengan quantile (nama, prob = c (.01, .03, ...))
5. Run coding.

Translation
1. Data input in R
2. Search for quartiles with quantile(name, prob = c(.25, .5, .75)
3. Finding deciles with quantile(name, prob = c(.1, .2, .3, ...)
4. Finding percentiles by quantile(name, prob = c(.01, .03, ...)
5. Run code

Figure 16. Pseudocode example of calculating quartiles, deciles, and percentiles

After successfully compiling the steps to calculate the size of the data location in R as presented in Figure 16, in the validation situation, the students tried to test the pseudocode directly in R. Finally, in the institutionalization situation, the students were challenged to solve a new problem, namely, calculating the size of more complex data locations in R on Google Colab. An example of the artifact of finding the size of the data location in R on Google Colab, produced by the students, is presented in Figure 17.

Figure 17 illustrates the process of finding quartiles, deciles, and percentiles using R in Google Colab in two steps. First, the students input data into R. Second, students search for quartiles, deciles, and percentiles using the "quantil" function. The difference between searching quartiles, deciles, and percentiles lies only in the probability. As a result, R presents the quartiles, deciles, and percentiles of the previously entered data.

The results of this study show that the didactical design developed has undergone a validation process by experts and has received various improvements based on their input. The implementation of the design in five learning tasks shows that this approach can integrate Computational Thinking (CT) into R-assisted descriptive statistics learning in Google Colab. Through a series of didactical situations-Action, Formulation, Validation, and Institutionalization-the students gradually built a conceptual understanding of statistics while developing computational thinking skills. In addition, the implementation process revealed how the students could adapt problem-solving strategies using R from the initial understanding stage to application in more complex data contexts. Each task was designed to hone the four main components of CT-Decomposition, Pattern Recognition, Abstraction, and Algorithm-so as to not only strengthen mastery of statistical material but also improve analytical skills and technology-based problem solving. These findings provide a strong foundation to further examine the effectiveness of this didactical design through retrospective analysis, which will be the focus of future research.

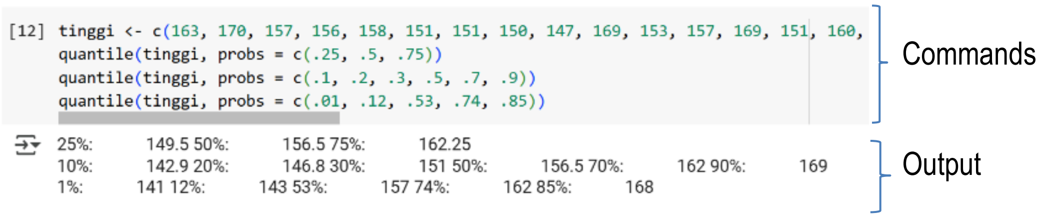


Figure 17. Artifact example of calculating quartile, decile, and percentile using R

Discussion

The primary aim of this research was to design a hypothetical didactical framework that integrates R software into a basic statistics course to enhance CT skills among undergraduate mathematics education students. This design targets third-semester students and is aligned with their foundational knowledge of statistics and introductory programming experience. The framework is grounded in the TDS (Brousseau, 2002) and incorporates essential didactical components to foster a productive learning environment.

The proposed framework adopts a three-dimensional structure (5 × 4 × 4), in which the five learning tasks—Task 1 (data, scales, variables, and R), Task 2 (data presentation), Task 3 (measures of central tendency), Task 4 (measures of dispersion), and Task 5 (measures of location)—are intersected by four didactical situations (action, formulation, validation, and institutionalization) and four CT components (decomposition, pattern recognition, abstraction, and algorithms) as identified by Dong et al. (2019). This design expands upon the earlier work of Piatti et al. (2022), who introduced a 3 × 3 × 3 cube integrating activity types, computational domains, and student autonomy. The current study extends this model by embedding more complex dimensions tailored specifically to the instructional needs of descriptive statistics using R in Google Colab, ensuring a more holistic integration of statistical reasoning and CT skills.

The hypothetical didactical design produced 80 distinct didactical situations distributed across five instructional sessions, with each session encompassing 16 didactical situations. These were derived from the combination of CT components and didactical stages, each accompanied by anticipated student responses and corresponding pedagogical interventions, referred to as Anticipation Didactique Pédagogique (ADP). The ADP accounts for two primary student pathways: one where students are able to solve the tasks independently, and another where students encounter difficulties. In the latter case, scaffolding strategies such as simplified prompts or guiding instructions are provided to facilitate self-directed problem solving (Anghileri, 2006; Bakker et al., 2015; Bell & Pape, 2012). Consequently, the

ADP functions both as an intervention tool for students experiencing cognitive obstacles and as a means of guiding incremental learning progression.

Each type of didactical situation within the Cuboid Framework serves a distinct pedagogical function. The Action Situation involves active student engagement in statistical exploration using R on Google Colab. During this phase, students are introduced to data manipulation and computation of various descriptive statistics, including measures of central tendency (mean, median, mode), dispersion (range, variance, standard deviation), and location (quartiles, deciles, percentiles). This exploratory phase supports the development of foundational statistical intuition, consistent with findings from Nurjanah et al. (2021), which emphasize the pedagogical value of hands-on experimentation.

In the Formulation Situation, students are encouraged to construct structured solutions by translating statistical problems into pseudocode and algorithmic representations before implementing them in R. This phase aligns closely with CT practices, particularly in fostering decomposition, pattern recognition, abstraction, and algorithmic thinking (Looi et al., 2018; Tareq & Yusof, 2024). The emphasis on procedural development reinforces logical sequencing and procedural fluency, critical for both statistical problem solving and programming competence.

The Validation Situation focuses on evaluating and verifying the functionality of students constructed solutions. By executing their code in R, students receive immediate feedback through output or error messages, prompting iterative debugging and refinement. This process mirrors authentic programming practices and strengthens diagnostic reasoning (Becker et al., 2016; Tareq & Yusof, 2024). The recursive cycle of testing and correction cultivates students' ability to identify logical inconsistencies and optimize algorithmic approaches.

In the Institutionalization Situation, students are required to generalize their learning by applying previously acquired knowledge to new, complex, and contextually relevant problems. These tasks demand not only statistical reasoning but also computational flexibility as students transition from guided to autonomous problem solving. Application contexts may vary from educational assessments to business analytics and social science research, thereby reinforcing the real-world relevance of statistical and computational integration.

The deliberate sequencing and integration of didactical situations within the Cuboid Framework have demonstrated potential to enhance both conceptual and procedural understanding. The ADP, as a forward-thinking instructional tool rooted in didactical theory (Suryadi, 2010; 2013), allows instructors to anticipate learning difficulties and respond adaptively. Its theoretical basis builds upon the refinement of the didactical triangle proposed by Kansanen and Meri (1999), emphasizing the dynamic interaction between teacher, student, and content.

Findings from this study indicate that the CT-integrated didactical design supports student engagement in statistically grounded problem solving and fosters CT development. The use of R software not only enhances students' statistical competencies but also provides an authentic computational environment for developing algorithmic thinking. Classroom observations and expert validation suggest that the structured implementation of didactical situations—particularly the validation phase—plays a crucial role in nurturing logical reasoning through debugging practices.

Overall, this study contributes a theoretically grounded and empirically supported framework for embedding CT into undergraduate statistics education. Future research should explore its long-term impact on students' statistical literacy and computational proficiency through experimental or longitudinal designs. Additionally, adaptation of this model to other mathematical topics and instructional technologies may broaden its applicability and pedagogical value.

CONCLUSION

This study successfully developed a didactical design framework for integrating CT into descriptive statistics learning, specifically for undergraduate mathematics education students. The Cuboid Framework systematically combines statistical tasks, didactical situations from the TDS, and CT components to enhance students' problem-solving abilities using R in Google Colab. Through expert validation and iterative refinement, this framework provides structured learning experiences that progressively develop students' conceptual understanding of statistics while fostering computational thinking skills. The findings indicate that the integration of CT into statistics learning through the Cuboid Framework effectively supports students in decomposing problems, recognizing patterns, abstracting information, and designing algorithms within statistical problem-solving contexts. Structured didactical situations—Action, Formulation, Validation, and Institutionalization—play a crucial role in guiding students through each phase of learning, ensuring a balance between conceptual exploration and computational application.

Despite its contributions, this study has several limitations. First, it focused primarily on the prospective analysis and metapedadidactic stages of Didactical Design Research (DDR), without extending to the retrospective analysis stage, which would involve a comparative quantitative evaluation of the framework's effectiveness in improving students' CT and statistical reasoning. Second, although expert validation confirmed the framework's theoretical alignment, its practical implementation across diverse educational settings remains unexplored. Future research should focus on empirical classroom implementation to assess the impact of the Cuboid Framework on student learning outcomes, including a quantitative analysis of its effectiveness. Additionally, adapting this framework to other mathematical and scientific domains could provide further insights into its generalizability and applicability beyond statistical learning. Continued refinement of this approach has the potential to enhance computational thinking integration in education, ensuring that students develop the necessary analytical and problem-solving skills for the data-driven era.

Acknowledgments

All authors would like to thank the experts who were willing to validate and provide valuable input on the hypothetical didactic design. In addition, our gratitude goes to the Tadris Mathematics study program at the Institut Agama Islam Negeri Ponorogo, especially to the students who took the Basic Statistics course in the 2023/2024 academic year.

Declarations

Author Contribution	:	EI: Conceptualization, Visualization, Writing - original draft, Methodology, and Writing - review & editing. RR: Supervision, Validation, Formal analysis, and Writing - review & editing. SP: Supervision, Validation, Methodology, and Writing - review & editing.
Funding Statement	:	This research was funded by the Director General of Strengthening Research and Development with the Ministry of Research, Technology and Higher Education of the Republic of Indonesia.
Conflict of Interest	:	The authors declare no conflict of interest.

Additional Information : Additional information is available for this paper.

REFERENCES

Abdul Hanid, M. F., Mohamad Said, M. N. H., Yahaya, N., & Abdullah, Z. (2022). Effects of augmented reality application integration with computational thinking in geometry topics. *Education and Information Technologies*, *27*(7), 9485–9521. https://doi.org/10.1007/s10639-022-10994-w

Acevedo-Borrega, J., Valverde-Berrocoso, J., & Garrido-Arroyo, M. del C. (2022). Computational thinking and educational technology: A scoping review of the literature. *Education Sciences*, *12*(1), 39. https://doi.org/10.3390/educsci12010039

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Adewumi, S. (2019). A systematic review of computational thinking approach for programming education in higher education institutions. *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, 1–10. https://doi.org/10.1145/3364510.3364521

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, *55*(7), 832–835. https://doi.org/10.1093/comjnl/bxs074

Akiba, D. (2022). Computational thinking and coding for young children: A hybrid approach to link unplugged and plugged activities. *Education Sciences*, *12*(11), 793. https://doi.org/10.3390/educsci12110793

Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International Journal of Child-Computer Interaction*, *19*(1), 30–55. https://doi.org/10.1016/j.ijcci.2018.10.004

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, *105*(106185), 1–8. https://doi.org/10.1016/j.chb.2019.106185

Anghileri, J. (2006). Scaffolding practices that enhance mathematics learning. *Journal of Mathematics Teacher Education*, *9*(1), 33–52. https://doi.org/10.1007/s10857-006-9005-9

Bakker, A., Smit, J., & Wegerif, R. (2015). Scaffolding and dialogic teaching in mathematics education: Introduction and review. *ZDM*, *47*(7), 1047–1065. https://doi.org/10.1007/s11858-015-0738-8

Becker, B. A., Glanville, G., Iwashima, R., McDonnell, C., Goslin, K., & Mooney, C. (2016). Effective compiler error message enhancement for novice programming students. *Computer Science Education*, *26*(2–3), 148–175. https://doi.org/10.1080/08993408.2016.1225464

Bell, C. V., & Pape, S. J. (2012). Scaffolding students' opportunities to learn mathematics through social interactions. *Mathematics Education Research Journal*, *24*(4), 423–445. https://doi.org/10.1007/s13394-012-0048-1

Benakli, N., Kostadinov, B., Satyanarayana, A., & Singh, S. (2017). Introducing computational thinking through hands-on projects using R with applications to calculus, probability and data analysis. *International Journal of Mathematical Education in Science and Technology*, *48*(3), 393–427. https://doi.org/10.1080/0020739X.2016.1254296

Brousseau, G. (2002). *Theory of didactical situations in mathematics* (N. Balacheff, Trans.). Kluwer Academic Publishers.

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, *109*, 162–175. https://doi.org/10.1016/j.compedu.2017.03.001

Chen, H. E., Sun, D., Hsu, T.-C., Yang, Y., & Sun, J. (2023). Visualising trends in computational thinking research from 2012 to 2021: A bibliometric analysis. *Thinking Skills and Creativity*, *47*(3), 1–18. https://doi.org/10.1016/j.tsc.2022.101224

Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology*, *28*(6), 651–676. https://doi.org/10.1007/s10956-019-09799-3

Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., & Andrews, A. (2019). PRADA: A practical model for integrating computational thinking in K-12 education. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 906–912. https://doi.org/10.1145/3287324.3287431

Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM*, *51*(8), 25–27. https://doi.org/10.1145/1378704.1378713

Haseski, H. I., Ilic, U., & Tugtekin, U. (2018). Defining a new 21st century skill-computational thinking: Concepts and trends. *International Education Studies*, *11*(4), 29. https://doi.org/10.5539/ies.v11n4p29

Ilic, U., Haseski, H. I., & Tugtekin, U. (2018). Publication trends over 10 years of computational thinking research. *Contemporary Educational Technology*, *9*(2), 131–153. https://doi.org/10.30935/cet.414798

Irawan, E. (2024). *Keterampilan computational thinking mahasiswa melalui penerapan desain didaktis dengan memanfaatkan perangkat lunak-R pada mata kuliah statistika* [PhD Thesis, Universitas Pendidikan Indonesia]. https://repository.upi.edu/

Irawan, E., & Herman, T. (2023). Trends in research on interconnection of mathematics and computational thinking. *AIP Conference Proceedings*, *2805*, 1–9. https://doi.org/10.1063/5.0148018

Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2024a). Advancing computational thinking in mathematics education: A systematic review of indonesian research landscape. *Jurnal Teori Dan Aplikasi Matematika*, *8*(1), 176–194. https://doi.org/10.31764/jtam.v8i1.17516

Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2024b). Harnessing the power of technology in statistics education: A comprehensive bibliometric study. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *58*(2), 108–126. https://doi.org/10.37934/araset.58.2.108126

Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2024c). Promoting computational thinking through programming trends, tools, and educational approaches: A systematic review. *Jurnal Teori Dan Aplikasi Matematika*, *8*(4), 1327–1348. https://doi.org/10.31764/jtam.v8i4.26407

Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2024d). Research trends of computational thinking in mathematics learning: A bibliometric analysis from 2009 to 2023. *Eurasia Journal of Mathematics, Science and Technology Education*, *20*(3), 1–16. https://doi.org/10.29333/ejmste/14343

Kansanen, P., & Meri, M. (1999). The didactic relation in the teaching-studying-learning process. *Didaktik/Fachdidaktik as Science (-S) of the Teaching Profession*, *2*(1), 107–116.

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, *72*, 558–569. https://doi.org/10.1016/j.chb.2017.01.005

Looi, C.-K., How, M.-L., Longkai, W., Seow, P., & Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. *Computer Science Education*, *28*(3), 255–279. https://doi.org/10.1080/08993408.2018.1533297

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of Computational Thinking, Digital Competence and 21st Century Skills When Learning Programming in K-9. *Education Inquiry*, *11*(1), 1–17. https://doi.org/10.1080/20004508.2019.1627844

Nurjanah, N., Dahlan, J. A., & Wibisono, Y. (2021). The effect of hands-on and computer-based learning activities on conceptual understanding and mathematical reasoning. *International Journal of Instruction*, *14*(1), 143–160. https://doi.org/10.29333/iji.2021.1419a

Palts, T., & Pedaste, M. (2020). A model for developing computational thinking skills. *Informatics in Education*, *19*(1), 113–128. https://doi.org/10.15388/infedu.2020.06

Piatti, A., Adorni, G., El-Hamamsy, L., Negrini, L., Assaf, D., Gambardella, L., & Mondada, F. (2022). The CT-cube: A framework for the design and the assessment of computational thinking activities. *Computers in Human Behavior Reports*, *5*(1), 1–56. https://doi.org/10.1016/j.chbr.2021.100166

Prahmana, R. C. I., Kusaka, S., Peni, N. R. N., Endo, H., Azhari, A., & Tanikawa, K. (2024). Cross-Cultural Insights on Computational Thinking in Geometry: Indonesian and Japanese Students' Perspectives. *Journal on Mathematics Education*, *15*(2), 613–638. https://doi.org/10.22342/jme.v15i2.pp613-638

Purwasih, R., Turmudi, & Dahlan, J. A. (2024). How do you solve number pattern problems through mathematical semiotics analysis and computational thinking? *Journal on Mathematics Education*, *15*(2), Article 2. https://doi.org/10.22342/jme.v15i2.pp403-430

Raykov, T., & Marcoulides, G. A. (2013). *Basic statistics: An introduction with R*. Rowman & Littlefield Publishers.

Rich, K. M., Franklin, D., Strickland, C., Isaacs, A., & Eatinger, D. (2022). A learning trajectory for variables based in computational thinking literature: Using levels of thinking to develop instruction. *Computer Science Education*, *32*(2), 213–234. https://doi.org/10.1080/08993408.2020.1866938

Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., & Franklin, D. (2018). K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals. *ACM Inroads*, *9*(1), 46–55. https://doi.org/10.1145/3183508

Roig-Vila, R., & Moreno-Isac, V. (2020). Computational thinking in education: Bibliometric and thematic analysis. *Revista de Educación a Distancia (RED)*, *20*(63), 1–24. https://doi.org/10.6018/red.402621

Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Computational thinking in high school science classrooms. *The Science Teacher*, *81*(05), 1–6. https://doi.org/10.2505/4/tst14_081_05_53

So, H.-J., Jong, M. S.-Y., & Liu, C.-C. (2020). Computational thinking education in the asian pacific region. *The Asia-Pacific Education Researcher*, *29*(1), 1–8. https://doi.org/10.1007/s40299-019-00494-w

Suryadi, D. (2010). Penelitian pembelajaran matematika untuk pembentukan karakter bangsa. *Prosiding Seminar Nasional Matematika dan Pendidikan Matematika UNY*, 1–14.

Suryadi, D. (2013). Didactical design research (DDR) to improve the teaching of mathematics. *Far East Journal of Mathematical Education*, *10*(1), 91–107.

Suryadi, D. (2019). *Landasan filosofis penelitian desain didaktis (DDR)*. Gapura Press.

Tang, K.-Y., Chou, T.-L., & Tsai, C.-C. (2020). A content analysis of computational thinking research: An international publication trends and research typology. *The Asia-Pacific Education Researcher*, *29*(1), 9–19. https://doi.org/10.1007/s40299-019-00442-8

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, *148*(1), 1–22. https://doi.org/10.1016/j.compedu.2019.103798

Tareq, Z.-A., & Yusof, R. J. R. (2024). Modeling a problem-solving approach through computational thinking for teaching programming. *IEEE Transactions on Education*. https://doi.org/10.1109/TE.2024.3354425

Tekdal, M. (2021). Trends and development in research on computational thinking. *Education and Information Technologies*, *26*(5), 6499–6529. https://doi.org/10.1007/s10639-021-10617-w

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

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

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, *25*(2), 7–14. https://doi.org/10.17471/2499-4324/922

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, *53*(4), 562–590. https://doi.org/10.1177/0735633115608444