Towards an integration of statistical and computational thinking

Development of a task design framework for introducing code-driven tools through statistical modelling

Anna-Marie Grace Fergusson

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in Statistics, The University of Auckland, 2022..

Abstract

The advent of data science has led to statistics education researchers re-thinking and expanding their ideas about what computational tools to use for teaching statistical modelling. Consequently, the use of computer programming languages such as R have been promoted in textbooks, tasks, and other learning activities to support data science at the high school level. The shift in pedagogical knowledge required for teachers, however, is not restricted to the introduction of code-driven tools and computational thinking to the curriculum. The statistical methods taught need modernising to include digital sources of data, such as APIs, and algorithmic approaches, for example, predictive and classification modelling. Therefore, teaching data science at the high school level using code-driven tools requires consideration of how learning activities can be designed to support the co-development of statistical and computational thinking.

Minimal research exists about the design of tasks that support the introduction of high school students and teachers to the use of code-driven tools for statistical modelling. Hence, the dual objectives of the study were to: (1) explore the observable thinking practices that emerge when learners completed statistical modelling tasks that introduced code-driven tools; and (2) develop a task design framework to introduce code-driven tools and support the integration of statistical and computational thinking. Using a design-based research approach, four structured tasks were developed for teaching statistical modelling at the same time as introducing the programming language R. These four tasks were implemented with high school statistics teachers across four full-day face-to-face professional development workshops.

The study resulted in the development of the Introducing Code-driven Tools (ICT) task design framework, which was produced by identifying key design elements for one task, and refining and re-evaluating design principles and processes through consecutive retrospective analyses of the other three tasks. The findings from this exploratory study indicate that the tasks constructed supported teachers' introduction to code-driven tools and encouraged an integration of statistical and computational thinking. The ICT task design framework contributes to statistics and data science education research by building on previous work in effective pedagogy, and by providing practical guidelines for the introduction of code-driven tools to facilitate the integration of statistical and computational thinking to learn from modern data. Another contribution is the production of two hypothesised frameworks, which provide guidelines for assessing and clarify the integration of statistical and computational thinking.

Acknowledgements

If it were not for Maxine Pfannkuch, I would never have embarked on this research journey. I see it as an unbelievable privilege that I have been able to learn from her and I am so appreciative of her patience and encouragement. Maxine has had a huge impact on my thinking and participation in statistics education, and I am a better researcher and teacher because of her guidance and support. Thank you for never giving up on me, and for teaching me so much about the importance of research for shaping teaching practice.

I am also hugely indebted to Chris Wild, for his unwavering championing of my teaching and technological innovations. Combined with the friendship and encouragement of Christine Franklin, our many conversations over many years have provided the necessary fuel for my research. Many thanks to Stephanie Budgett and Pip Arnold, for always knowing the right things to say at the right time. Through Maxine, Chris, Christine, Stephanie, and Pip, I have been privileged to meet educators from around the world. In particular, I want to thank the wider SRTL research community for their friendship, encouragement, and valuable critique of my research.

This research would not have been possible without the teachers who gave up many Saturdays to participate. Teaching is a challenging and complex job, and I am so grateful that you offered to share your insights and experience with me. I would also like to thank my students for their enthusiasm and honesty about learning data science, and my HOD James Curran for supporting my study and encouraging my research and teaching.

Although this thesis is ”mine”, I would not have been able to complete it without the support of others. To Anne Patel, Michelle Dalrymple, Emma Lehrke, Mark Hooper, and Lars Thomsen - thank you for your endless encouragement, provocations, and understanding, particularly when I struggled to do ”all the things.” Thank you to my Mum for instilling into me a love for education, and to my Dad for giving me my first book on computer programming.

The hugest thanks of all goes to my best friend and husband Leo. Thank you for everything, and I mean everything. These last five years have been challenging for many reasons but you have been unwavering in your encouragement. I owe you so much, and not just in terms of housework (I know, no more excuses)! Words can not express how much your love and support has meant to me.

Table of contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Motivation

Towards the end of 2015 I attended a talk at a national teachers conference and keynote speaker Rob Gould challenged the audience with the statement:

> Why do we teach statistics? Because of data! Don't select data because of its mathematical properties, teach what is needed to analyse and understand the data students interact with every day of their lives.

At that time, I was writing new teaching materials for a very large introductory level statistics course and constructing tasks that explored modern data contexts (see Fergusson & Bolton, 2018). Rob's words resonated strongly with me, as *following the data* was a key focus for my development of these tasks. A few years later, I began my PhD study at the same time as I began designing a new introductory level data science course. As I continued to construct, implement, and refine learning tasks for teaching data exploration and statistical modelling, it became apparent that in order to use "the data students interact with every day of their lives" (Gould, 2015), I needed to re-think my task design approach. My knowledge of data technologies such as web scraping, databases, APIs, and computer programming languages was a crucial and important factor that enabled me to imagine, design, and craft new learning activities. I realised that the tasks I developed would not be the same as the ones I had crafted previously over my 14-year statistics teaching career. To teach students how to understand the data they interact with through a range of digital technologies would require integrating statistical and computational thinking.

It did not seem to me that using a computer programming language was as easy as just switching from using "point and click" software tools to script-based code approaches. Instead, I had to think carefully about *when* to use code to teach statistical ideas, *how much* code to use to teach statistical ideas, *how much* “raw” code to reveal to students versus using functions that hid many of the computations, and *how much* understanding about the code itself needed to be taught versus focusing in the output of the code. I looked for examples being shared of innovative modern approaches to teaching statistics using computer programming. I became frustrated, however, that the focus was on *what* to teach and *what* tools to use, or the provision of *what* activities to use and *what* technology to use to create them. The focus was not on *how* to create effective tasks and *why* these tasks would support the development of statistical and computational thinking. That is, teachers were not being supported to create their own tasks, as an explication of *how* to create their own tasks was not given. Furthermore, no research evidence was provided that the tasks were supporting learners to integrate statistical and computational thinking.

Cobb once stated that none of the key innovations in the undergraduate statistics curriculum were due to statistics education research (Cobb, 2015b). Instead, he agreed with Utts (2015) that textbooks were one of the main drivers of substantial change in statistics education and that the role of statistics education research was to *document* “whether and in what ways existing approaches do or do not help students learn” (Cobb, 2015b, p. 5). Although I agree with Cobb that statistics education research plays a crucial role in exploring *how* pedagogical approaches support students to learn, I do not agree with Cobb’s distinction between innovation and research in statistics education. Drawing on my experience as Head of Department for Mathematics and Statistics at a very large co-educational high school, I know that textbooks *can* provide examples of good teaching tasks, but they very often do not provide the guidance needed to equip teachers to create their own tasks, particularly when technology is involved.

When I create tasks, I use my many years of designing and implementing learning tasks, my reading about effective pedagogy and my beliefs about how to introduce new knowledge to learners. This is not unusual approach for task designers and researchers. As Arnold et al. (2017) remarked, implicit design principles are often incorporated into learning activities and more consideration is needed in this area. Explicating the implicit was the driver for research by Wild and Pfannkuch (1999), as they claimed that they knew good statistical thinking when they saw it but were unable to explicate and characterise it for other people. In a similar vein, I believed my tasks for introducing teachers to code-driven tools through statistical modelling would be effective in promoting learning, which led to the realisation that my PhD research should focus on extracting the design principles behind my tasks and at the same time provide evidence that the approach helped students to learn. Such a focus would provide teachers practical guidance on how to introduce high school students to code-driven tools and create their own tasks.

Therefore, within the context of teaching statistical modelling at the senior high school level, my research will explore design principles for tasks that introduce code-driven tools and provide evidence that the tasks help learners to integrate statistical and computational thinking. The rest of this chapter will provide a brief background to this research topic, outline the rationale for the research, identify the key research questions, define the scope of the research, and summarise how the thesis has been structured.

1.2 Background

The advent of data science has led to statistics education researchers re-thinking and expanding their ideas about tasks and tools for teaching and learning. A common thread to discussions about data science education is that students need to integrate both statistical and computational thinking to learn from data (e.g., De Veaux et al., 2017), which necessitates students developing at least some coding (computer programming) skills (e.g., Gould, 2010). Using code-driven tools facilitates the use of digital data sources and algorithmic modelling, which may support the integration of statistical and computational thinking. Although most of the current recommendations for data science education are framed within the context of tertiary education (e.g., Cetinkaya-Rundel & Rundel, 2018), introducing code-driven tools within high school statistics classrooms is also consistent with the digital technology goals of schools.

The implementation of the 2007 New Zealand statistics curriculum (Ministry of Education, 2007) has already had a significant impact on the nature of the statistics taught and assessed in secondary schools (Forbes et al., 2014). The curriculum changes not only placed a greater emphasis on statistical modelling and thinking within a statistical enquiry cycle, but also defined statistics as a connected but separate subject from mathematics at the senior high school level. New approaches to statistical modelling, such as simulation-based methods, were introduced into the curriculum, and new computational tools were developed to support the teaching and learning of these simulation-based methods. One of the enablers of freeing statistics from the chains of mathematics (Cobb, 2015a) was the increased affordability, availability and usability of computers and other digital technologies. Within statistics education research and practice, computational tools such as *TinkerPlots* (Konold & Miller, 2015), the *Common Online Data Analysis Platform* (CODAP, Finzer, 2016), *iNZight* (Wild, 2018), and *Visual Inference Tools*

Online (VIT Online, Wild & Halstead, n.d.) have gained popularity, due to their careful design and use of animation, graphics and interactivity to support development of statistical concepts, thinking, and reasoning.

When introducing coding as part of data science at the high school level, careful considerations are needed to ensure that the development of statistical concepts, thinking, and reasoning are still well supported when learners engage with code-driven tools. One consideration must be task design, as tools by themselves do not teach (Wild, 2018). The use of tools is paired with tasks, and the task used will influence the nature of the learning that takes place (Doerr & Pratt, 2008). However, task design research in mathematics education has only recently gained attention (Watson & Ohtani, 2015). Within statistics education, research focused on task design has not specifically explored computer programming as the main computational tool (e.g., Ben-Zvi et al., 2017), although design principles have been developed for dynamic interactive mathematics technologies (e.g., Burrill, 2016; Dick & Burrill, 2016). Any task design principles for teaching mathematics and statistics using GUI-driven tools, however, cannot be assumed to automatically apply to construction of tasks for data science. Consideration of task design research from within computer science is also required, as well as research that connects data technologies and pedagogy.

1.3 Rationale for research

It is my belief that data science education at the senior high school level should involve the use of code-driven tools. As computer programming is not required as part of the New Zealand school curriculum for statistics, it cannot be assumed that high school statistics teachers have a good knowledge of coding, nor knowledge of how to design statistical modelling tasks that use code-driven tools. For high school teachers to be confident data science teachers, they need to co-develop skills in *doing* statistical modelling using code-driven tools, and *constructing tasks* for teaching statistical modelling using code-driven tools. For these reasons, and because my research is exploratory, I decided to focus my research on high school statistics teachers. The purpose of this research is to explicate design principles for the construction of statistical modelling tasks that introduce code-driven tools. The dual objectives of the study are to: (1) describe the observable thinking practices that emerge when learners completed tasks that introduced code-driven tools within the teaching and learning context of statistical modelling and; (2)

develop a task design framework, comprising a cohesive set of design principles and processes, to guide construction of such tasks.

1.4 Research questions

The main and supporting research questions are:

1. What observable thinking practices emerge as teachers, positioned as learners, engage with statistical modelling tasks that introduce code-driven tools?
    - Can these observable thinking practices be characterised as integrating statistical and computational thinking? and if so,
    - What features of the tasks appear to stimulate or support the integration of statistical and computational thinking?
2. What design principles could guide the construction of statistical modelling tasks that introduce code-driven tools?
    - How could tasks be constructed to support the introduction of new sources of data and modelling approaches, simultaneously with new code-driven tools?
    - Does using familiar computational tools or modelling approaches within the same task support the introduction of new code-driven tools?

1.5 Scope of the research

An exploratory study on task design principles for statistical modelling from a data science perspective was conducted over five years using a design-based research approach (e.g., Bakker & van Eerde, 2014). My analysis of practical teaching issues and construction of new learning tasks was informed by both existing design principles and technological innovations (e.g., Edelson, 2002; Reeves, 2007). As there was limited research literature for task design specifically in the area of statistical modelling with code-driven tools, relevant task design literature from mathematics and computer science was also considered.

The participants in the research were twelve teachers (nine female and three male) with experience teaching the national statistics curriculum for the last two years of high school. The high school statistics teachers were positioned as learners for the research. I constructed four new tasks that were aligned to the national curriculum for statistics at the year 12 or 13 level (the last two years of high school). The curriculum focus was statistical modelling from a data science perspective. The tasks extended the following curriculum topics: numeric data distributions, probability simulations, simple linear regression, randomisation (permutation) tests. All the tasks introduced the programming language *R* (R Core Team, 2020).

1.6 Thesis structure

Chapter 1 provides the motivation for researching the design tasks that introduce code-driven tools through statistical modelling, at the senior high school level. Chapter 2 reviews education research about what statistical modelling tasks and computational tools should be used, from the perspective of data science. After establishing the need to create tasks that support the integration of statistical and computational thinking, the chapter presents potential guidelines for statistical modelling task design that introduces code-driven tools. Chapter 3 describes the research methods used, and explains how a pragmatic design-based research approach was used to guide the task construction, data collection, and analysis methods. The results for the research, including the development of the ICT (Introducing Code-driven Tools) task design framework, are developed over Chapters 4, 5, 6 and 7. Chapter 8 presents the final version of the ICT task design framework and reviews relevant existing theories and principles for task design from mathematics, statistics, and computer science education. Chapter 9 discusses the ICT task design framework, contributions to the research knowledge base, implications of the framework for teaching and research, and limitations of the research and task design framework.

Chapter 2: Background

2.1 Introduction

The review of literature started when I began to look for research and ideas to inform the implementation of data science at the senior high school level. In the recommendations for what to teach, I saw there was little exploration in the literature of the potential for blending different types of computational tools within the same learning programme or task, for example, using both GUI-driven and code-driven tools for teaching data science. I initially formed a working hypothesis that when introducing code-driven tools, teachers should include and build on experiences with familiar tools, rather than replacing GUI-driven tools with code-driven tools for teaching statistical modelling. To explore this possibility, I developed and implemented tasks that introduced code-driven tools, drawing on characteristics of simulation-based modelling activities developed for tools such as *VIT Online* (Wild & Halstead, n.d.) and *TinkerPlots* (Konold & Miller, 2015).

The scope of Chapter 2 is a review of the literature that was available during 2017 to 2019. During this period, I concurrently developed and implemented tasks for my own teaching at the tertiary introductory level and developed tasks for my research at the senior high school level. The focus of the chapter is intentionally practical and reflects the duality of my roles as teacher and researcher. Recommendations and relevant research for teaching data science at the senior high school are reviewed in Section 2.2, with particular attention given to the situated teaching context of Aotearoa New Zealand, the expanding sources of data, and the teaching of statistical modelling. A summary is then provided in Section 2.3 of teaching implications related to thinking frameworks and computational tools, and Section 2.4 summarises how both of these are connected with task design. The chapter ends with potential guidelines for statistical modelling task design involving code-driven tools in Section 2.5, identification of gaps in the literature, and a re-statement of the research questions.

2.2 Teaching data science at the senior high school

The rapid development of modern data and associated digital technologies requires an urgent review of what is taught at the senior high school level. What it means to learn from data is different because of advances in computing (Baumer et al., 2017) and consequently students need to learn new computational skills to support them to understand and interpret modern data (Gould, 2015; Nolan & Temple Lang, 2010). Although there are an increasing number of commentaries about the need for data science education at the high school level (e.g., Engel, 2017; Finzer, 2013; Ridgway, 2015), there has been very little research into teaching and learning of data science at this level. Furthermore, there is no consensus view on how data science should be implemented at the senior high school level. While at the tertiary and industry level, there may be acceptance that data science involves some mix of statistics and large-scale computing (Biehler, 2018; Hoerl et al., 2014), there is no agreement on the nature and scale of the computational techniques or tools required to implement data science at the senior high school level. The specification of these computationally-bound techniques and tools is crucial for any professional development work with upskilling high school teachers on how to teach students to learn from modern data.

2.2.1 Defining data science as a curriculum subject

Definitions of data science are deeply connected to the author's motivations for providing such definitions. I have adopted a broad view that data science involves extracting meaning from data that has been sourced from our modern digital world, and that computational knowledge, like contextual knowledge, shapes how we use and how we think about data and models. As I will elaborate further in this chapter, I contend that teaching data science requires a substantial shift in how learning activities are designed. It is my view that any data science curriculum needs to be developed with the use of computational tools as a central component throughout (Hicks & Irizarry, 2018), rather than viewing data science as adding to, modernising, or enhancing an existing statistics curriculum and its teaching and assessment practices. Although I agree that the statistical ideas and concepts taught at the senior high school level should be enduring in the face of ongoing changes in technology (Pfannkuch, 2018), students' learning of statistical concepts and methods will be limited if they are not taught specific computational skills to enable them to learn from modern data (c.f. Cleveland, 2001). To support these learning goals, I think it is important that students at the senior high school level develop sufficient computer programming skills to allow them to be active participants in the science of learning from data, which includes being able to create computational products. Consequently, I have taken the position that data science is not the same as statistics, and that the amount of change that is necessary is so substantial that it would be advisable and strategic to use data science as a new name to describe the desired curriculum subject at the senior high school level (c.f. Cleveland, 2001).

Not all agree that data science at the senior high school level should be conceived of as one subject taught by teachers situated within one department (e.g., Finzer, 2013). While data is indeed not the exclusive domain of statistics, a similar argument could be made about mathematics. For example, when arguing for computational thinking to be specifically taught as part of the digital technologies (computer science) curriculum in Aotearoa New Zealand, Bell (2016) reasoned:

> So, the relationship of computational thinking to the curriculum is a bit like that of maths or English; you could argue that these needn't be taught in their own right because they would be used by other subjects anyway, but at some point, you need to acknowledge that there are some valuable concepts that students might not encounter by chance, and ensure that they are covered.

According to De Veaux et al. (2017), students need to co-develop statistical and computational thinking, and I argue that data science being implemented as one subject at the senior high school level could provide learning experiences that support this co-development. Data science could be a subject taken alongside statistics, mathematics or computer science. Because some universities present data science as a combination of statistics and computer science courses at the undergraduate level, it could be imagined that implementation at high school could involve combining different existing statistics and computer science topics and be co-taught by high school statistics and computer science teachers. However, Sun (2018) argues that dedicated data science courses are needed to provide the "glue that fills the gaps" and that statisticians should be the ones teaching them. The relationship between statistics and data science education is evident, but the amount of commonality depends on the country or state. To understand the Aotearoa New Zealand school system within which my research is situated some background knowledge of the New Zealand statistics curriculum, associated teaching and assessment practices, and country-specific influences on curriculum change is now discussed.

In Aotearoa New Zealand, statistics is taught across all year levels and can be taken as a subject in its own right in the last two years of high school (called Year 12 and Year 13). In 2007, the national curriculum was re-written, and re-named from Mathematics to Mathematics and Statistics (Ministry of Education, 1992; Ministry of Education, 2007). Year 13 students can gain entry to tertiary-level study using Mathematics or Statistics as one (or two) of their three required approved subjects. Like many other countries, senior level statistics courses typically use data collected within formal studies such as surveys and experiments to teach students about study design, statistical inference and models. However, unlike other countries who seek greater space and time for the teaching of statistics within mathematics, statistics is already a significant component of the senior high school curriculum, due in large part to the long history of statistics education research undertaken within Aotearoa New Zealand (Forbes, 2014; Hipkins, 2014). For example, calls to adopt a modelling perspective for the teaching of statistics (e.g., Garfield et al., 2012) and probability (e.g., Pfannkuch & Ziedins, 2014) have been listened to through making stronger connections between data and chance (e.g., Konold & Kazak, 2008) and taking advantage of computing (e.g., Cobb 2007; Nolan & Temple Lang, 2010).

New approaches to statistical modelling, such as simulation-based methods, were introduced into the curriculum, and new computational tools were developed to support the teaching and learning of these simulation-based methods. One of the enablers of freeing statistics from the constraints of mathematics has been the increased affordability, availability and usability of computers and other digital technologies (Cobb, 2015). With respect to the assessment of the national curriculum, the highest level of statistical thinking is characterised as "an integration of statistical and contextual knowledge throughout the statistical enquiry cycle", supported by other practices including: informed contextual knowledge; reflecting about the process; discussing how possible sources of variation were dealt with during the design phase; considering other relevant variables; considering other relevant explanations; evaluating the adequacy of any models; and showing a deeper understanding of models (Starnes & Martin, 2015). Some of the existing teaching and assessment practices supported by the current statistics curriculum include: students doing real investigations with real data; the use of multivariate data sets; students posing investigative questions from a data set; wide-spread adoption of technology to facilitate learning from data; the specific teaching and assessment of statistical literacy; and development of student understanding of inference using visual inference tools; and probability and probability modelling.

The statistics curriculum in Aotearoa is very extensive and there is no room to introduce data science from the perspective I have described, one which places modern data and practical skills with digital technologies at the core. Additionally, positioning data science as a subject implemented at the senior high school level is consistent with how statistics was itself introduced as a subject into the New Zealand curriculum. As Begg (2004, p. 1) explained:

> Being introduced at senior level enabled one or two teachers in most high schools to become familiar with statistics. Later statistics became part of mathematics at all levels of high school, and in the next round of curriculum change it became part of mathematics for all students from age five. The success of these initiatives was partly due to the time frame and to starting with a small group of able teachers.

To further define what data science as a subject at the high school could look like, I now review literature to identify content and pedagogical approaches that are new or different from the current statistics curriculum in Aotearoa New Zealand.

2.2.2 Identification of curriculum content

For students to be able to work with modern data, broader knowledge of and practical skills with digital technologies are needed. Advances in computing power require new approaches to teach students how to learn from data (Baumer et al., 2017). The amount, availability, diversity and complexity of modern data requires educators to broaden their definitions of what data is and what it means to teach students how to learn from data (Finzer, 2013; Gould, 2010). Teaching activities should use data sourced from the data revolution (Ridgway, 2016) and include learning experiences that reveal more of the world of data faster (Wild, 2015).

Looking across the literature for data science curriculum content (e.g., American Statistical Association, 2014; Biehler & Schulte, 2017; Baumer et al, 2017; Cobb, 2015a; De Veaux et al., 2017; Engel, 2017; Fergusson & Bolton, 2018; Finzer, 2013; Gould, 2010; Gould, 2017; Hardin, 2018; Hicks & Irizarry, 2018; Horton et al., 2014; Kaplan, 2018; Kim et al., 2018; NASEM, 2018; Ridgway, 2016; Sun, 2018), key content were identified. This content was then compared with four examples of data science high school curricula, to refine the selection of curriculum content: Bootstrap Data Science (BDS, bootstrapworld.org/materials/data-science), International Data Science in Schools Project (IDSSP, idssp.org/pages/framework.html), Introduction to Data Science (IDS, idsucla.org), and ProDaBi (prodabi.de). Content that did not feature strongly in the current statistics curriculum for Aotearoa New Zealand was selected. The curriculum content identified can be grouped into two broad categories: (1) accessing and creating data using computational tools; (2) developing statistical products, such as models and visualisations, using computational tools. The curriculum content for data science should include:

Accessing and creating data using computational tools

- Sourcing static and dynamic data using modern data-related methods: web scraping, querying databases, and interacting with Application Programming Interfaces (APIs)
- Creating data from a wide range of sources: text, images, sounds, movements, sensors, social media, interactions with digital devices, and participatory activities
- Reading and storing unstructured and structured data, including rectangular (tidy), hierarchical, spatial, and file types such as CSV, XML, JSON
- Wrangling and processing “messy” data for a purpose, for example, creating new categorical variables, creating aggregate views for particular group, or automating the identification of “suspicious” values
- Accessing and using data from open sources: government statistics, open data websites, data repositories
- Data-related quality issues: sources of bias, sparsity, considerations of generalisability
- Considering the relationship between people and data: ownership, sovereignty, ethics, privacy, civic responsibilities, social implications of data use

Developing statistical products, such as models and visualisations, using computational tools

- Exploring and communicating with data for purposes other than sample-to-population inference: creating infographics and other visually-based reports, finding structure for machine learning, sentiment analysis
- Creating a wide range of types of data visualisations from large complex data sets: visualising three or more variables, producing spatio-temporal graphics such as geo mapping
- Modelling relationships between observations not just relationships between variables: algorithmic approaches such as clustering and natural language processing (NLP)
- Machine learning approaches: prediction and classification models such as decision trees, training and testing data sets, model validation
- Simulation-based modelling: learner-defined probability simulations for inference, utilising 1000s of trials with computational tools, creative projects such as generative art
- Reproducible data analysis: computational essays, automated reporting systems
- Using programming languages and associated functions to develop, use, and express models

All the curriculum content identified involves at least some knowledge of statistical, computational, and mathematical approaches. The challenge then becomes: How much and at what level do these different approaches get combined? Recommendations for what data science to teach do not include age-appropriate progressions or consider how statistical concepts may be developed across different curriculum levels. Students in Aotearoa New Zealand are likely to have been exposed to key statistical concepts and approaches before reaching the senior high school level and this familiarity has implications for teaching data science. On the one hand, extending familiar statistical approaches into unfamiliar data science approaches has been recommended as a strategy for teaching (Biehler & Schulte, 2017). On the other hand, there is a potential for confusion when complementary but different approaches to learning from data are combined within the same learning programme.

One implication of the curriculum content identified is that algorithmic modelling approaches may be taught alongside traditional modelling approaches. Brieman (2001) refers to these two approaches as "two cultures" for how statistical modelling is used to reach conclusions from data. One culture assumes that the data generating process can be modelled by a probability model, whereas the other culture makes no assumptions about the data generating process and consequently uses algorithmic models. Educators have proposed that learning from, and blending aspects, from both "cultures" is a good basis for teaching data science (Grant, 2017; Sun, 2018). Both probabilistic and algorithmic approaches for statistical modelling use data, and so students will need to consider "traditional" statistical issues such as assumptions, biases, and the different sources of uncertainty (Grant, 2017). With respect to modelling approaches, students will need to extend their understanding beyond the traditional use of linear regression, to focus on new perspectives such as the use of training and testing data, residual analysis, predictive accuracy, overfitting models, and cross validation (Biehler & Schulte, 2017; Grant, 2017). Additionally, algorithmic models could offer a more accessible and conceptually simpler mechanism to introduce students to data science than inferential methods (Baumer et al., 2017; Gould, 2017; Grant, 2017; Ridgway, 2016), and support an expansion or redefinition of what it means to teach Exploratory Data Analysis (EDA).

Broadening the nature of the data used for teaching data science means that the data used for statistical modelling may not be randomly sampled and so the use of probability models for observation or opportunistic data is not necessarily relevant (Baumer et al., 2017; Gould, 2010). Furthermore, the traditional approach of refining ill-defined research questions for the purpose of making sample-to-population or experiment-to-causation inferences may no longer apply. Instead, students will need to extend their repertoire to consider, for example, criteria for questions based on the kinds of answers they provide: descriptive, exploratory, inferential, predictive, causal, and mechanistic (Peng & Matsui, 2017). For a data science course that combines traditional elements with newer approaches such as machine learning, classifying questions based on what they allow you to do with data, and what type of modelling you will undertake, appears to be a helpful approach at the senior level (cf. Arnold, 2013). Data considerations also need to expand beyond the use of statistical knowledge. In Aotearoa New Zealand, for example, data science education needs to include and promote Matauranga Māori and Te Ao Māori approaches to data.

The issue of how to deal with mathematics when teaching statistics has already been identified by educators such as Cobb (2015a) as one of the biggest issues in statistics education, with tensions "between mathematics and data, between abstraction and context, between theory and story" (p. 2). Although aspects of statistical modelling, including algorithmic modelling, require abstraction (Pfannkuch, 2011), there should be minimal reliance on mathematical notation (Hicks & Irizarry, 2018). Researchers caution against using "black box" approaches to teaching modelling (e.g., Biehler & Schulte, 2017; Grant, 2017; Magana et al., 2011) and therefore attention is needed on how the task design highlights and promotes key statistical concepts (Bargagliotti & Groth, 2016) and on the development of sequences of tasks. I now consider pertinent literature pertaining to the development of statistical modelling-based activities.

2.2.3 Developing statistical modelling-based activities

Because the implementation of a data science curriculum will involve combining concepts and methods from statistical and computational spheres, a focus on statistical modelling could provide an entry point into the world of data science, particularly if prediction models such as classification are used. Given earlier discussion, it will be important to develop learning activities that help students to understand the "two worlds" of statistical modelling: probability-based and algorithm-based. Design principles for tasks developed from research concerning probabilistic reasoning by Pfannkuch and Budgett (2016) suggest that making conjectures, testing conjectures, linking representations and relatable contexts are important features of statistical modelling activities. Further guidance for designing and structuring modelling activities can be found in the literature associated with Model Eliciting Activities.

Model Eliciting Activities (MEAs) are activities that encourage students to invent, test, and communicate about models as part of investigating an open-ended problem (Lesh et al., 2000). MEAs support students to reveal, refine and extend their thinking as they work in small groups to provide a model-based solution to a provided problem. MEAs also provide a way for teachers to learn more about students' thinking and so inform teaching practice (Lesh et al., 2003). Six design principles guide the construction of MEAs: (1) reality, (2) model construction, (3) self-assessment, (4) construct documentation, (5) construct share ability and reusability, and (6) effective prototype (Lesh, et al., 2000). Because a key learning goal for MEAs is that students develop a conceptual model that is general enough to be used across a range of contexts and situations (Lesh et al., 2000; Patel & Pfannkuch, 2018), sequences of model development activities are needed rather than just the use of single MEAs in isolation (Ärlebäck et al., 2013).

Model development sequences are structurally related tasks that begin with a MEA and are followed by other modelling activities (Lesh et al., 2003). MEAs and model development sequences are intended to develop conceptual models that students recognise as being important as well as practical (Lesh et al., 2000), particularly if students complete the learning sequence by solving a problem that would have been too difficult to solve before (Lesh et al., 2003). Some researchers refer to the activities that are used after MEAs as model exploration activities, model adaptation activities, model extension activities and model application activities (e.g., Ärlebäck et al., 2013; Lee et al., 2016; Lesh et al., 2003), and others refer to these as follow-up tasks or follow-up activities (e.g., Patel & Pfannkuch, 2018; Radonich, 2014; Yoon et al., 2011). Lee et al. (2016) describe *MEAs* as revealing students' initial understanding of how to model a situation, *model exploration activities* as focusing students on the underlying structure of the model, and *model application activities* as requiring students to apply the model to new contexts. Lesh et al. (2003) further distinguish between MEAs and model exploration activities by explaining that model exploration activities can involve the use of computers and should aim to develop strong language and representation systems related to the model.

MEAs and model development sequences were originally developed within the context of mathematics education but are used in other areas of education such as statistics and engineering (e.g., Garfield et al., 2010; Hjalmarson et al., 2008; Lee et al., 2016; Patel & Pfannkuch, 2018). When MEAs are used within statistics education literature, the activities are typically designed to elicit students' understanding of specific statistical ideas or methods rather than mathematical models. Garfield et al. (2010) explained that when they developed materials for the CATALST (Change Agents for Teaching and Learning Statistics) project, they created a new type of MEA to elicit statistical thinking with models, with the following three requirements (p. 2):

1. Reflects a real type of statistical problem (e.g., making predictions using multiple variables or figuring out a way to classify objects to make predictions, based on known data)
2. Has a current and engaging context that will motivate students to work on a solution and also illustrates the relevance of statistics to their everyday lives
3. Uses real data either gathered in a research study or gathered for the purpose of the MEA

Similar adaptations when using MEAs for teaching statistics can be seen in model development sequences used by other researchers (e.g., Lee et al., 2016; Patel & Pfannkuch, 2018).

Although MEAs use open-ended problems they are not unstructured tasks. Explicitly, MEAs typically involve four components: a hook, for example, a newspaper article or short video; warm-up questions to assess readiness; data or other relevant mathematical information that needs to be used for the activity; and a problem statement involving a client, which provides the motivation for a model-based solution (Chamberlin & Moon, 2005; Patel & Pfannkuch, 2018). Implicitly, the activity should be designed so that the structure reveals itself naturally as students complete the task rather than the structure being “forced” by the teacher (Lesh et al., 2000).

To understand the task design Garfield et al. (2010) used implicitly for two of their MEA tasks as part of the CATALST project, I analysed them from a design perspective. My analysis revealed that their tasks had teacher-led phases that appeared to be important in developing statistical modelling ideas for both probability models and algorithmic models. For instance, both tasks guide the students through building then testing their models. Both tasks also provoke students to develop methods to evaluate their models or use their models to produce results. Figure 2.1 presents my naming and characterisation of the different structural phases used in the iPod shuffle and spam email MEAs described by Garfield et al. (2010). Similar structural features can be seen in the MEAs and model development sequences created for probability modelling (e.g., Lee et al., 2016; Patel & Pfannkuch, 2018).

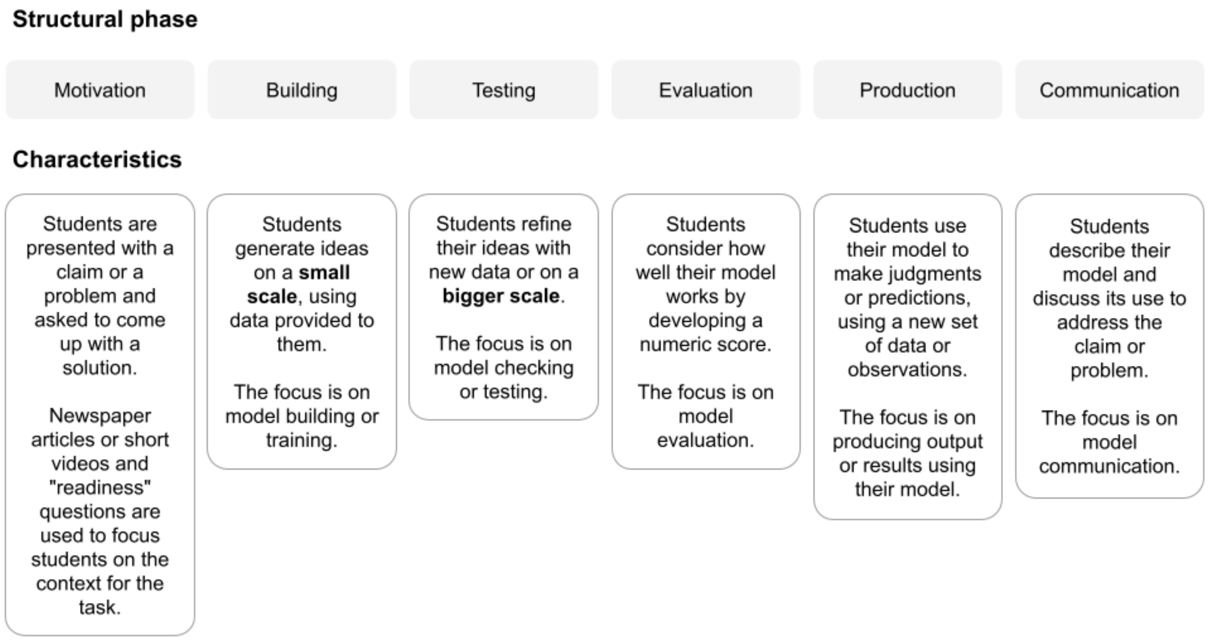

Structural phase
Motivation
Building
Testing
Evaluation
Production
Communication
Characteristics
Students are presented with a claim or a problem and asked to come up with a solution. Newspaper articles or short videos and "readiness" questions are used to focus students on the context for the task.
Students generate ideas on a **small scale**, using data provided to them. The focus is on model building or training.
Students refine their ideas with new data or on a **bigger scale**. The focus is on model checking or testing.
Students consider how well their model works by developing a numeric score. The focus is on model evaluation.
Students use their model to make judgments or predictions, using a new set of data or observations. The focus is on producing output or results using their model.
Students describe their model and discuss its use to address the claim or problem. The focus is on model communication.

Figure 2.1: Structural phases used in iPod shuffle and SPAM email MEAs

My analysis of the two MEA tasks shows that the same structural phases appear to be present when eliciting modelling ideas related to hypothesis testing using a probability-based model (iPod shuffle) and classification (spam email) using an algorithm-based model. For this reason, I conjecture that using statistical model development sequences may be a promising approach for creating learning tasks for teaching data science at the high school level, that support the "two cultures" of statistical modelling (cf. Breiman, 2001). However, the literature reviewed thus far does not specifically account for how teachers can develop their students' thinking when using digital technologies or tools, for example, computational thinking. Therefore, the literature review now turns to inherent thinking practices in data science and focuses on statistical thinking and computational thinking, and their integration.

2.3 Thinking practices for data science

The data students interact with in their everyday lives is predominantly digital (Bell & Roberts, 2016; Gould, 2010) and could provide rich opportunities for students to deeply explore contexts that matter to them (cf. Weiland, 2016). Additionally, investigations involving modern data could support the development of critical thinking and scepticism (cf. Hicks & Irizarry, 2018) and highlight the role of human decision making in the data analysis process (De Veaux & Velleman, 2015). Using computational tools, including reading and writing computer instructions (coding), requires new computational ways of thinking about data and modelling to be used alongside statistical thinking (De Veaux et al., 2017). To teach data science, teacher guidance is needed for how to develop learning activities that promote both statistical and computational thinking practices, and how these activities may be different from those familiar to high school statistics teachers. Statistical and computational thinking and related frameworks are now reviewed, to explore possible ways of integrating statistical and computational thinking.

2.3.1 Statistical thinking and related frameworks

The term "statistical thinking" was originally used by statisticians such as Snee (1990) and Moore (1990) to highlight the need to consider, and account for, variation throughout all aspects of learning from data. Rather than emphasising "statistical rituals" (Gigerenzer, 1998) and mechanical procedures, statistical thinking should instead draw on core statistical concepts to inform decision making in the face of uncertainty (De Veaux et al., 2017). Statistical thinking can be considered as a set of desired thinking practices that support the understanding of why and how statistical investigations are conducted (Ben-Zvi & Garfield, 2004). From an inquiry perspective, statistical thinking goes beyond individual components of statistical reasoning and literacy, and requires thinking about the whole statistical process (Chance, 2002; Hardin et al., 2015).

To support teaching, Wild and Pfannkuch (1999) developed a four-dimensional framework to characterise statistical thinking. The first dimension of this framework, the investigative cycle, and to a lesser degree the third dimension (the interrogative cycle), have been used extensively by statistics education researchers to inform the design and analysis of learning activities (e.g., Leavy & Hourigan, 2016; Pfannkuch et al., 2011). For instance, Arnold (2013) undertook a comprehensive inquiry into statistical investigations, leading to her theory of statistical questions and a set of frameworks for supporting the teaching of statistical thinking. Dimensions two and four of the Wild and Pfannkuch's (1999) statistical thinking framework have received less attention in the research literature. Dimension two of the framework called "types of thinking" describes five elements that are fundamental to statistical thinking (p. 226). These are:

1. Recognition of the need for data
2. Transnumeration: changing representations to engender understanding
3. Consideration of variation
4. Reasoning with statistical models

5. Integrating statistical and contextual knowledge

All five of these elements are entirely relevant to learning from modern data. For instance, learning to recognise when data is needed could include becoming more aware of data generated algorithmically (Gould, 2010). Converting unstructured text into a rectangular data set or filtering a data set (Erickson et al., 2019) will require transnumeration. Exploring randomness, variability, and uncertainty could involve using large scale simulation-based methods, enabling the creation and visualisation of synthetic data sets (Hardin et al., 2015). Modern data contexts also provide ample opportunities for students to explore variation, model variation, and evaluate models (cf. Stigler & Son, 2018). Statistical thinking is required when using algorithmic modelling with large data sets, in order to develop a sound strategy for understanding and reducing variation (Hoerl et al., 2014).

With respect to integrating statistical and contextual knowledge, exploiting modern data contexts can support development of important research questions, motivate students to learn, and assist development of technical understanding (e.g., American Statistical Association, 2014; Cobb, 2015a). Moreover, modern data contexts may also be more engaging for teaching high school students (Gould, 2010; Ridgway, 2016). Issues such as algorithmic bias and data ethics could be leveraged to highlight the subjectivity of analytical and modelling processes, including building an awareness that an overuse of personal contextual beliefs may inhibit learners from applying statistical methods appropriately. For example, Patitsas et al. (2019) found in a randomised experiment involving computer science professors that professors were more likely to describe distributions of grades as bimodal, if they were first primed to think about the commonly-held belief that computer science grade distributions are bimodal. Although the context of the data can help learners make meaning from what they see in the data (Wild & Pfannkuch, 1999), relying too much on contextual explanations can get in the way of building generalisations and statistical concepts (Pfannkuch, 2011).

In the fourth dimension of their framework, Wild and Pfannkuch (1999) propose a set of dispositions for thinking statistically. Their dispositions for statistical thinking include: scepticism, imagination, curiosity and awareness, being logical, and perseverance. Other educators have also recommended more focus on encouraging students to think more personally about their connections and actions with data, such as developing data habits of mind (Finzer, 2013). When considering the role of statistical thinking in teaching data science, criteria based on dispositions could provide a more flexible means for assessing student learning, particularly when considering the role of computing. For example, Wickham (2010), developed a rubric for grading data analysis tasks where two of the six key criteria are scepticism and curiosity. Another important related consideration is the use of frameworks to guide teaching and learning of data science.

The four examples of data science high school curriculum referred to earlier in this literature review use different frameworks to support the teaching of statistical thinking. The Introduction to Data Science (IDS), Bootstrap Data Science (BDS), and International Data Science In Schools Project (IDSSP) curricula use frameworks that are similar in structure to the PPDAC (Problem, Plan, Data, Analysis, Conclusion) cycle, the first dimension of Wild & Pfannkuch's (1999) statistical thinking framework. The IDS and BDS use the "Data Cycle" which consists of four steps: (1) Pose questions; (2) Consider data; (3) Analyse data; (4) Interpret data. All these four steps are based on a specific research topic. None of the frameworks used by IDS, BDS, and IDSSP specify modelling as one of the steps or phases, nor do they appear to include the *creation of computational products* that might result from an inquiry cycle.

In contrast, the ProDaBi curriculum uses the CRISP-DM (CRoss-Industry Standard Process for Data Mining) framework. The CRISP-DM framework places data at the core of the process, and the connected core components are: business understanding, data understanding, data preparation, modelling, evaluation, and deployment. The CRISP-DM framework does include modelling as a core component, and also explicitly includes the creation of computational products through its deployment component. Similarly, drawing on data practices employed by data scientists, RStudio developed the "Data science workflow" that comprises the following: import data; tidy data; understand data by iteratively visualising, transforming, and modelling; infer understanding of data to other data sets; and communicate results to others and/or automate analysis for re-use. Notably, the "Data Science Workflow" specifies that computer programming is used throughout and includes the creation of computational products.

A key limitation for all these frameworks for teaching data science is that they were not developed as frameworks for constructing tasks. The frameworks instead capture and explicate key aspects of the processes that should be used by data scientists to learn from data (cf. Hicks & Irizarry, 2018). Although education researchers do not expect students engaging in statistical investigations to follow the steps of inquiry frameworks in the order given (Gould et al., 2017), teaching tasks used by teachers are often constructed using these steps. Further research is needed to explore how learners *actually* move between different phases of frameworks and how these pathways may support or hinder statistical thinking.

Gould et al. (2017), for instance, explored how two pairs of high school teachers made decisions as they analysed data and aligned their actions to each step of the "Data Cycle". They found the teachers used a variety of pathways between the different "Data Cycle" steps. Importantly, both pairs of teachers used questioning extensively as they progressed through their investigations, and the pair of teachers who spent more time raising questions and crafting productive questions appeared to be more successful. However, investigating non-traditional data that has a computational basis, such as the participatory sensing data used by Gould et al., requires additional knowledge and thinking beyond the statistical and contextual. Asking questions of modern data requires computational thinking.

2.3.2 Computational thinking and related frameworks

Computational thinking is a relatively new educational goal compared to statistical thinking. Made popular in recent years by Wing (2006), definitions of computational thinking vary and depend on the intended teaching context. For the purposes of teaching data science, it is important to identify *relevant* aspects of computational thinking that may provide students with the necessary critical thinking skills for learning from modern data, beyond what statistical thinking can already provide. According to Wing (2006, p. 33), computational thinking involves "solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science." Nardelli (2019, p. 34) provided a more general definition that computational thinking "is the thought processes involved in modelling a situation and specifying the ways an information-processing agent can effectively operate within it to reach an externally specified (set of) goal(s)." Lee et al. (2011) proposed that computational thinking requires abstraction, automation, and analysis.

Due to the increased popularity of integrating computational thinking across the school curriculum, various frameworks for computational thinking have been developed. For example, a three-dimensional framework for computational thinking was developed by Brennan and Resnick (2012), based on their observations of how students interacted with the Scratch programming language. The three dimensions of their framework are concepts, practices, and perspectives. The computational thinking concepts described in their framework are specifically tied to programming constructs and include: sequences, loops, parallelism, events, conditionals, operators, and data (storing, retrieving and updating values). The four key computational thinking practices are experimenting and iterating, testing and debugging, reusing and remixing, and abstracting and modularising. Perspectives needed for computational thinking included realising that computation is a medium of creation, appreciating the value of creating products with and for others, and being empowered to use computation to ask and answer questions about the world.

Although Brennan and Resnick's computational thinking framework was developed based on students' interactions with a computer programming language, it is important to note that computational thinking is not the same as coding (computer programming). Computational thinking not only involves learning practical skills with computers but also includes developing conceptual understanding of how computational processes happen (Wing, 2006). Furthermore, a range of tools can be used for computational thinking, not just programming languages (Repenning et al., 2010). One reason why computational thinking is often seen as synonymous with coding is that writing and executing code is used as an efficient way to test the soundness of a student's computational thinking (Bell & Roberts, 2016). Additionally, learning programming allows student to use computers to *create* digital technologies rather than just *use* computational tools.

There does not appear to be a framework specifically for teaching computational thinking alongside statistical thinking. The closest existing framework appears to be the taxonomy created by Weintrop et al. (2015) to define computational thinking for mathematics and science classrooms. The researchers used computational thinking literature, interviews with mathematicians and scientists, and existing computational thinking teaching materials to devise a set of four key practices: data practices, modelling and simulation practices, computational problem-solving practices, and systems thinking practices. Each of the four practices contains up to seven individual computational thinking practices, resulting in 22 different computational thinking practices. However, it is difficult to imagine how such a framework could help data science teachers at the senior high school level design tasks that integrate statistical and computational thinking, as the taxonomy is not conceived from a statistical or modern data perspective.

These frameworks do not provide guidance for how specific computational tools, such as computer programming languages, can be introduced within other subject domains. Research involving the use of Model Eliciting Activities (MEAs) to teach mathematical thinking alongside computational thinking indicated that designing effective tasks was a complex process. For example, Shoop et al. (2016) developed a MEA that used robotics activities to teach proportional reasoning. The researchers noted that it took four iterations to find a version of the MEA that reliably improved mathematics skills. One version put too much focus on mathematics, and another version placed too much focus on the robotics context. Extending these results to designing a data science task involving statistical modelling and coding, it is conceivable that an activity might provide the necessary experiences to help students understand procedural computations, such as use of specific functions, but may not provide sufficient statistical conceptual learning opportunities. Therefore, it will be important to target key computational thinking practices that support students to learn from modern data and integrate these with statistical thinking practices.

2.3.3 Integrating statistical and computational thinking

Despite calls for data science curricula to provide opportunities for students to integrate both computational and statistical thinking (e.g., De Veaux et al., 2017; NASEM, 2018), there are few research-based examples of what such an integration could look like at the senior high school level. Some examples are recommendations that students develop at least some computer programming skills (e.g., Cetinkaya-Rundel & Rundel, 2018; Gould, 2010; Nolan & Temple Lang, 2010), whereas other examples are of statistical modelling exercises that use computer programming in curriculum materials. As discussed, computational thinking is not the same as computer programming and while coding provides one way to develop and demonstrate computational thinking, there are other ways that need to be explicated for prospective teachers of data science.

According to the Cambridge dictionary, *to integrate* means “to combine two or more things in order to become more effective” and synonyms include connect, combine, and blend. An immediate challenge with turning “integrating statistical and computational thinking” into a practical task or activity is that the options for “combining” are infinite. Integrating statistical and computational thinking does not require that both types of thinking are stimulated or observed for every aspect of a learning activity (cf. Finzer, 2013). Nor does an integration require that the “blend” of statistical and computational thinking is equal. At a minimum, a selective approach is needed about what aspects of computational thinking are the most applicable for integrating with statistical thinking at the senior high school level. For instance, the IDS curriculum (Gould et al., 2017, p. 11), states that for computational thinking, “students will learn to write code to enhance analyses of data, to break large problems into smaller pieces, and to understand and employ algorithms to solve problems.”

Interpreting “integrating” as “connecting” may provide a more feasible approach to teaching the integration of statistical and computational thinking. Wild and Pfannkuch (1999) propose that statistical thinking requires shuttling between the contextual and statistical spheres. The contextual sphere provides the questions for data, and the answers need to be investigated within the statistical sphere. However, to interpret the features seen in the data and provide meaningful answers to the initial questions requires shuttling back to the contextual sphere. By this reasoning, learners are never located in both spheres at the same time, but learn from one to inform the other, constantly moving between both spheres to create understanding when learning from data. Therefore, integrating statistical and computational thinking could involve connecting and shuttling between three spheres: the contextual, the statistical, and the computational.

Knowing more about the computational world of data and algorithms will support students to ask and answer a wider range of data-related questions, such as: *How do we find the data? How do we retrieve the data? How can we assess the quality of the data? What variables do we need to derive from the raw data? How can the data be organised into a structure for analysis? Will the approach remain computationally feasible as the number of documents increases?* (Hardin et al., 2015, p. 344). As Gelman and Nolan explain (2017, p. 342):

> When we teach statistics, we stress that the context of the problem matters, that the context needs to be considered when analysing data to answer a question or provide a solution to a problem. Relatedly, we think computational considerations are part of this context because an understanding of how we might analyse the data can impact how we process the data, and reciprocally, considerations about the size and the organisation of the data can impact our analysis.

Case studies that demonstrate connecting contextual, statistical, and computational spheres could be useful for identifying examples of observable thinking practices. In an interview with Alan Rossman for the Journal of Statistics Education (Rossman & Nolan, 2015), Deb Nolan used a case study she wrote with Danny Kaplan (see Nolan & Temple Lang, 2015) to explain computational thinking in the context of analysing data. The case study itself is 55 pages long and explores modelling runners' times in the "Cherry Blossom" race, covering statistical and computational challenges such as: *How to scrape data from the web? How to check if the data is what you expected or wanted? How to visualise tens of thousands of results meaningfully? How to work with time conversions? How to restructure data?* Although such case studies do provide practical examples of computational thinking concepts being applied within modern data contexts, they do not provide teaching strategies for supporting students to integrate statistical and computational thinking, nor do they demonstrate what integrated statistical and computational thinking could look like without the use of a code-driven computational tool.

For an integration of statistical and computational thinking to be taught and assessed, the practices associated with the integrated thinking process need to be described in observable terms. Such descriptions could support teachers to design specific learning activities to promote data scientific thinking and to use students' responses and actions to evaluate observed thinking practices. However, it is difficult to find examples within the statistics education research literature that specifically explain how different features of a task support development of specific components of statistical thinking. It is more common to find guidance for constructing tasks that are based on different phases of the statistical enquiry (e.g., Tran & Lee, 2015) but these often assume the use of non-code-driven tools. Assessment databases created as part of projects such as LOCUS (Jacobbe et al., 2014) and ARTIST (Garfield & delMas, 2010) provide valuable examples of questions and criteria for assessing aspects of statistical thinking but do not necessarily consider a computational dimension. Conversely, examples of assessing computational thinking at the school level assume the use of computer programming as the tool and contexts such as game or multimedia design (e.g., Brennan & Resnick, 2012; Hoover et al., 2016). To further explore how statistical and computational thinking, and their integration could be taught and to look for relevant guidance for teaching statistical modelling within a data science subject at the senior high school level, a review of the literature related to use of computational tools from the perspective of task design is conducted.

2.4 Tools for teaching data science

In considering the implementation of data science as a subject at high school level, it is yet to be resolved what kinds of computational tools students could use to learn from data. A common thread to discussions about data science education is that students need to integrate both statistical and computational thinking, and that this necessitates students developing at least some computer programming skills (Cetinkaya-Rundel & Rundel, 2018; Horton et al., 2014; Nolan & Temple Lang, 2010). Within the context of high school education, teaching programming is also consistent with the digital technology goals of today's schools. Any tool used for teaching data science needs to be considered from a learning perspective, including the nature of the tasks developed, how the tool and task together stimulate an integration of statistical and computational thinking, and the design of the tools themselves. Hence, after using simulation-based statistical modelling as a context for how computational tools could be used for teaching data science at the high school level, relevant statistics and computer science education research is reviewed for the purpose of informing the design of statistical modelling tasks that introduce computer programming.

2.4.1 Simulation-based statistical modelling

There are a variety of computational tools available for teaching data science at the senior high school level and a variety of ways in which these tools could be used. In general terms, computational tools can be described as interactive digital systems designed for a two-way flow of information between the user and the tool. Based on user actions such as *text input*, *a gesture, point, or click*, or *a voice command*, the computational tool responds, and this response influences further actions for the user. I will describe computational tools which users interact with predominantly by entering and executing text commands or scripts as *code-driven*, and computational tools which users interact with predominantly by pointing, clicking, or gesturing within visual environments as *GUI-driven* (Graphical User Interface-driven). I have not explored computational tools which users interact with predominantly through using voice commands. I used the word "driven" carefully for this research, to emphasise how the user is "driving" the interactions with the computational tool.

Computational tools for teaching data science include statistical software packages, graphics calculators, spreadsheets, programming languages, "microworlds" or custom-built applications, interactive tutorials or books, and interactive documents such as notebooks (cf. Ben-Zvi, 2000; Biehler, 1997a). Advice for selecting tools to teach statistics has included that the selection of tools should be in terms of learning goals and that a combination of tools may be best for student learning (e.g., Chance et al., 2007; Hesterberg, 1998). Statistics education researchers have carefully considered how computational tools influence students' thinking when learning statistics and have also considered how to define the complex relationship between software features, task design and learners' statistical conceptions (e.g., Ben-Zvi, 2000; Biehler, 1997b; delMas, 1997). It is important that computational tools are used to focus on developing statistical concepts (Chance & Rossman, 2006) and that consideration is given to how learners simultaneously develop conceptual-based and tool-based understanding (cf. instrumental genesis, Artigue, 2002).

An example of the crucial consideration of the relationship between tool, task and thinking

(Biehler, 2018; Doerr & Pratt, 2008; Moore, 1997) is the statistics education research concerning simulation-based inference and probability modelling, or simulation-based statistical modelling. The last decade has seen growth in the use of simulation-based statistical modelling within introductory data science courses. Simulation-based statistical modelling involves replacing a real or assumed random situation with a model, which can be manipulated and used to generate data that can be analysed (Engel, 2010). Technology-enabled simulations have been used to introduce students to probability and statistical inference since the earliest days of computers in classrooms (e.g., Simon et al., 1976; Thomas & Moore, 1980), but advances in digital technologies have enabled the development of intuitive tools for supporting and visualising results (e.g., Ben-Zvi, 2000). Consequently, software tools developed or used by statistics education researchers for teaching simulation-based statistical modelling have been predominantly GUI-driven (e.g., Applets: Chance & Rossman, 2006; Tools: Lock et al., 2013; Wild, 2018; *TinkerPlots*: Garfield et al., 2012; Noll et al., 2018) and the tasks developed using these computational tools are intertwined with the design of the tools.

There is some empirical evidence that simulation-based inference implemented with GUI-driven tools has a positive influence on learning (e.g., Beckman et al., 2017; Chance et al., 2016; Garfield et al., 2012; Tintle et al., 2012), for instance, a greater appreciation of the probabilistic nature of inference (e.g., Tintle et al., 2018). The use of simulation also promotes a modelling perspective to learning (Garfield et al., 2012) and gives students the opportunity to construct their own knowledge of statistics (Gelman & Nolan, 2017). When using technology for probability modelling, learners can quickly run multiple simulations and explore the relationship between the model and the data it generates (e.g., Pfannkuch & Budget, 2016), tinkering with a model and visualising changes instantaneously. Ideally, the computational tools used to carry out simulations would place the focus on defining the model, not the physical or time-consuming tasks of generating each trial manually. In this way, technology is used as a mechanism to shift statistics from being focused on computations, formulae and procedures towards a learning culture that supports informal and open exploration of data and models with minimal barriers (e.g., Ben-Zvi, 2000; Wild, 2018).

Research, however, has also highlighted aspects of statistical modelling that remain challenging for students (e.g., Case et al., 2018). Students and teachers can confuse real distributions with simulated distributions (Gould et al., 2010; Matuszewski, 2018; Pfannkuch et al., 2014). This suggests that simulated data need to be treated differently from observed data, especially when teachers also need to consider how simulation introduces another source of variation for the data (Engel, 2010; Watkins et al., 2014). Hence, there is a need to define data as "real" or "model-generated" more clearly, in line with several proposed statistical modelling frameworks (e.g., Case & Jacobbe, 2018; Fergusson, 2017). Although the use of simulation-based inference can make statistical inference reasoning more accessible to students, the nature of argumentation and hypothesis testing ideas remain difficult for students (e.g., Case et al., 2019; Pfannkuch et al., 2013). Because there are a large number of multiple images of representations that students are expected to grasp (Pfannkuch et al., 2013) when learning simulation-based probability modelling, GUI-driven tools are often selected by task designers due to the visual and interactive learning environment they provide (Forbes et al., 2014). However, Wild et al. (2017, p. 22) cautioned that "visualisation software does not teach - learning aided by visualisation software is a teacher-mediated process" and suggested that teachers ask students to explain and describe the key elements of what they are doing with the tool.

Researchers recommend that both unstructured and structured learning activities are used for teaching simulation-based statistical modelling (Chance et al., 2004; Engel, 2010; Gould et al., 2010). An unstructured approach could involve giving students a framework to carry out an investigation and then asking students to describe reasons for their conclusions, without a set of procedural steps (Gould et al., 2010). A structured approach could involve starting with a hands-on simulation, creating a link between the hands-on simulation and the computer-based simulation, and then moving to a computational tool to carry out the same simulation (Chance & Rossman, 2006). Similarly, delMas (1997) recommends that teachers provide clear examples of how models are built and interpreted and provide physical activities as well as computer simulations. Using "unplugged" tools at the start of the learning activity supports students to use physical actions to represent steps in the modelling process (Wood, 2005), so that their understanding can be connected to what the computer is doing (e.g., Chance et al., 2004; Erickson, 2006; Gould et al., 2010; Pfannkuch et al., 2013). Technology that provides "virtual" versions of physical chance devices such as coins and spinners has also been described as critical, so that students view the computer models they build as generating the same data that they would obtain using the actual devices (Konold & Kazak, 2008). Unplugged approaches include using data cards to create visual representations (e.g., Arnold et al., 2011) or shuffling cards by hand to simulate random allocation of treatments to units (e.g., Budgett et al., 2013).

Simulation-based statistical modelling can minimise mathematics being a barrier to learning formal statistical inference (Forbes et al., 2014; Ricketts & Berry, 1994; Thomas & Moore; 1980; Wild et al., 2017), and could provide a meaningful way to utilise computer programming within a data science curriculum (e.g., Cetinkaya-Rundel & Rundel, 2018; Horton et al., 2014; Nolan & Temple Lang, 2010). However, there is a potential tension with respect to using code-driven tools for teaching data science at the senior high school level. Statistics has only recently emerged as a separate subject from mathematics at the high school level and one of the enablers has been the increased affordability, availability and usability of GUI-driven tools. GUI-driven tools are often used as a mechanism to shift statistics from being focused on computations, formulae and procedures towards a learning culture that supports informal and open exploration of data and models with minimal barriers (e.g., Ben-Zvi, 2000; Wild, 2018). Therefore, I now review the literature to determine what pedagogical considerations are needed for using code-driven tools to teach statistical modelling.

2.4.2 Pedagogical considerations for using code-driven tools

Although statistics education research has predominantly focused on the use of GUI-driven tools for teaching simulation-based statistical modelling, code-driven tools have been used to support learners to understand probability and inference since the first days of computers in classrooms (e.g., Simon et al., 1976; Thomas & Moore, 1980). When teaching simulation-based statistical modelling, the use of a coding approach has the potential to allow for exploration of "what if?" scenarios, explorations that are often restricted by the options provided by GUI-driven tools. For instance, Ferreira et al. (2014) found that when high school students were introduced to the programming language *R* for conducting simulations, they quickly became familiar with the tool and experienced a sense of control in being able to manipulate the code to change their simulation output. Kaplan (2018) proposed that the use of code-driven tools allows students to demonstrate decision-making as part of statistical thinking. However, there are a number of decisions that may need to be made when developing a task using a code-driven tool, including: which computer programming language to use; the environment within which to read, write and execute code; the syntax and layout of the code used, including what packages to use if any; how much of the code to reveal or how much to "hide" within functions; and whether there is a need to write new functions.

An example of a code-driven tool currently used at the high school level for teaching statistics is the graphics calculator. In particular, the web-based graphics calculator DESMOS, which has been adopted by assessment organisations including within Aotearoa New Zealand. It is noteworthy that graphics calculators are not necessarily used because teachers believe they are the best for teaching statistical concepts but because they can also be used for teaching mathematics (Biehler, 2018). When learners interact with a graphics calculator by inputting text, the experience is similar to writing and executing computer code using the command line interface. It is therefore understandable that teachers, who do not currently teach statistical modelling using graphics calculators, may similarly view the use of code-driven tools as reverting to the less intuitive and more mathematical tools for learning statistics. A related concern is that teaching time will be diverted away from data analysis with GUI-driven tools to coding lessons (Finzer & Reichsman, 2018). However, the benefit of using any code-driven tool for teaching statistical modelling cannot be viewed in isolation from how it is used within a task, and the design of the tools themselves also requires pedagogical focus (Pratt et al., 2006). In the case of the graphics calculator, design principles for its use need to be considered when developing tasks (e.g., Burrill, 2017).

Additionally, when using code-driven tools, consideration needs to be given to the language and notation used to express computational actions or models (cf. Kaplan, 2007). Statistical computing educators have developed packages for different programming languages to support novices to learn statistics through the careful design of functions and consideration of syntax. Some examples of packages developed for the programming language *R* include the *mosaic* package (Pruim et al., 2017), the *infer* package (Bray et al., 2018) and the *mobilizr* package (Molyneux et al., 2017). A specific goal for the development of the *mosaic* package was that the functions provided would focus learner attention to the important parts of the statistical method and hide the details not relevant for learning (Pruim et al., 2017). The approach of writing "wrapper" functions, high-level functions that use other functions and obscure some of the underlying processes being used, has been encouraged by educators as a way to reduce the number of "computational templates" needed when transferring thinking to a code-driven tool (Grolemund & Wickham, 2014). As Hesterberg (1998, p. 7) described, "An instructor may use graphical interfaces and high-level functions to hide many details of computations from students, but some students will learn more from programming simulations themselves." Therefore, the consideration of what "to hide" and what "to reveal" computationally is important and can be referred to as *computational transparency.*

By *computational transparency*, I refer to how obvious the computations performed by the code are to the learner (cf. JamaicaVM). My conception is similar to how the term *transparency* is used within mathematics education, where Heid (1997, p. 6) defines transparency as "... the extent to which the technology being used highlights the mathematics that is being studied rather than obscures it." There appears to be no research that examines, within the context of statistical modelling, what level of *computational transparency* is needed when introducing code-driven tools. I propose that *computational transparency* is an entirely relevant pedagogical consideration and is strongly related to the claim that using code to express ideas, for example, describing models, will assist student learning. Educators have argued that code can be used as a tool for communication (Gelman & Nolan, 2017; Wickham, 2018), and that by using code to express modelling steps the cognitive demands of the statistical modelling task may be lowered (Kaplan, 2007). Chaput et al. (2008) proposed that the learning benefit from using code-driven tools for simulation-based statistical modelling is that learners need to analyse the random situation to describe a model for the situation with code. Research has also found that using code to express a model can help to articulate different theoretical perspectives on probability (Abrahamson et al., 2006). However, before learners can ask, (1) What do I want the computer to do for me? and (2) What does it need to know in order to do that? (Pruim et al., 2017), learners need to understand the statistical problem they want the computer to help them solve. That is, before learners can use code to express ideas, they need to have some prior experience with the idea before using the code-driven tool.

When using code-driven tools for teaching data science, consideration is needed about how both the tool and task support learners to access statistical concepts graphically (Ben-Zvi, 2000). However, this can pose a challenge for novices, as the mental models needed to produce graphics using code-driven tools are different from those used for generating data from models. Learners may require less computational transparency and greater scaffolding and support when using code-driven approaches to produce graphics within the same task. A challenge for teaching statistical modelling using code-driven tools is not just *how to use code* to express computational actions but also *how much* computer programming to teach. As Gould et al. (2018, p. 425) have recommended:

> While obtaining some level of proficiency with particular technological skills is important for completing the task at hand, having a conceptual framework and the confidence necessary to continue to learn new data technologies as required is even more valuable, especially since it will never be possible to teach students all the computational skills that would be beneficial in the context of a statistics program.

A pedagogical strategy to assist learners to engage with code-driven tools, while simultaneously providing structure and support for new computational ideas, is to provide students with interactive documents such as RMarkdown files (Allaire et al., 2018). RMarkdown files can be used as "computational templates", providing the code needed to import or generate data, analyse data, visualise data, and communicate about data (e.g., Hardin, 2018). Used within an interface such as RStudio (RStudio Team, 2018), students are also able to use GUI-driven actions to view the data, which is important as the data being used in the tool should be transparent to learners (Cobb & McClain, 2004). However, recommendations to use interactive documents such as RMarkdown are not likely to be widely adopted at the high school level as they involve installing and using specialist software. An alternative approach could be to use the R package *learnr* (Schloerke et al., 2018) to create interactive tutorials in which students execute small "chunks" of R code within a web browser. Similar approaches have been used to construct web-based textbooks for introductory level statistics.

In the past, there has been criticism of students using web applications specifically designed to support the learning of specific statistical concepts (e.g., Biehler, 1997a; Cetinkaya-Rundel & Rundel, 2018; Hesterberg, 1998; McNamara, 2015; Nolan & Temple Lang, 2007) and this criticism could be extended to the learning of specific computational concepts. Additionally, high school teachers will need to know where to find existing and relevant interactive tutorials, and have the pedagogical and computational knowledge to adapt the task or create new tasks of their own. An advantage of using an interactive tutorial environment for introducing computer programming is that the desired interactions between the tool and the learner's statistical and computational thinking can be embedded within both the tool and task design (cf. closed microworlds, Biehler, 1997a). Hence, a review of relevant statistics and computer science education research is needed to inform the design of statistical modelling tasks that introduce computer programming.

2.4.3 Introducing code-driven tools

There is minimal research from a statistics education perspective that has either explored pedagogical strategies such as task design for introducing code-driven tools for teaching statistical modelling. One reason is that data science education research is a relatively new field, but another reason is that GUI-driven tools dominate statistics education research (see Ben-Zvi et al., 2018). In considering how to introduce code-driven tools for teaching data science, educators have been advised to look to the computer science education community for pedagogical approaches (Gould et al., 2018; McNamara, 2015). A potential challenge with heeding this advice is that the resources available for computer science education are substantial and not all aspects of pedagogy and research may be applicable for crafting tasks that seek to develop statistical thinking with data. As discussed in Section 2.3.2, computational thinking frameworks can cover many practices, and not all of these should be a focus for teaching data science at the high school level. There is also the challenge of attempting to co-develop two related but different ways of thinking: statistical and computational thinking.

Research with high school statistics students, where the computer programming language *R* was used alongside other computational tools, revealed that the process of using a code-driven tool to develop statistical and computational knowledge is complex. For example, when encountering issues with executing code, learners often framed problems as either statistical or computational, and struggled to make connections between their knowledge of the computational tool and their knowledge of statistics (Thoma et al., 2018). Students may also hold different beliefs about what is important when using a code-driven tool for a statistical learning task, such as prioritising getting the code to work versus understanding the statistical concept (Dietrick et al., 2017). Students may also struggle to make connections between what they do with code-driven tools and what they do in classroom activities, with teachers playing an important role in encouraging these connections as part of classroom discussion (personal communication, Gould, 2018). Educators should also consider whether existing pedagogical approaches within statistics education developed with GUI-driven tools could be adapted for use with code-driven tools.

Pedagogical approaches used for teaching simulation-based statistical modelling using GUI-driven tools, or code-driven tools such as graphics calculators, may be relevant for introducing computer programming to data science students at the senior high school level. Using "unplugged" or hands-on activities first before moving to computational tools is a pedagogical approach also used within computer science education (Sentance & Csizmadia, 2017). *Computer Science Unplugged* (CSU) refers to activities designed to engage students with computational thinking, that is, computer science concepts rather than programming skills. One of the key principles of the unplugged approach is to address the barrier that programming can present for both students and their teachers (Bell et al., 2009). Bell said in an interview (2016, p. 5) that the benefits of using this approach extended to empowering teachers, as "they already know how to work with cards, string and chalk, and how to teach young children, so it provides the glue for them to do something without having to worry about digital devices crashing or being incompatible with the school system."

CSU activities are similar to the MEAs used for teaching statistical modelling (Garfield et al., 2012). For example, in CSU activities, students could work together to build models for computational solutions, and present these models to the class for analysis, discussion and optimisation before developing code (Shoop et al., 2016). Importantly, this means that development of *computer science concepts* happens before use of a code-driven tool. Computer science education researchers have also explored using interactive GUI-driven tools to develop understanding of programming concepts before moving to code-driven tools (Grover et al., 2019). Therefore, an approach where data science students learn statistical concepts using unplugged and GUI-driven tools first before moving to code-driven tools is consistent with computer science education research. Furthermore, developers of the ProDaBi data science curriculum used a similar approach, where *CODAP* was used to introduce high school students to exploratory data analysis before moving to the computer programming language Python via a Jupyter notebook (Biehler, 2018).

Another computer science teaching strategy identified by Sentance and Csizmadia (2017) in their research with high school teachers was the need to scaffold and structure programming tasks, which is also recommended by other computer science educators (Lee et al., 2011; Repenning et al., 2010). The PRIMM framework developed by Sentance et al. (2019) explicates a structured approach to developing a learning task for computer programming, based on the learning actions Predict, Run, Investigate, Modify, and Make. Similar to Gravemeijer's (2004) recommendation that statistics students meet the informal first before the formal, the PRIMM framework aims to shape student thinking and conceptual understanding over a progression of learning experiences. A key pedagogical strategy implicit in the PRIMM framework is that students start by reading code and predicting what the code might produce, consistent with longstanding research within computer science education that focusing on reading code before writing is helpful for programming novices (Van Merrienboer & Krammer, 1987). More recent research also suggests that when students are learning computer programming, they could benefit by describing and reading code aloud (Hermans et al., 2018).

In summary, a common feature of tasks that seek to develop either statistical or computational thinking is the use of "unplugged" activities before moving to computational tools. While code-driven tools may be a natural alignment to the computational process of statistical modelling, GUI-driven tools may offer greater access and support for statistical concepts. To summarise my initial considerations of thinking and tools, I developed the framework shown in Figure 2.2.

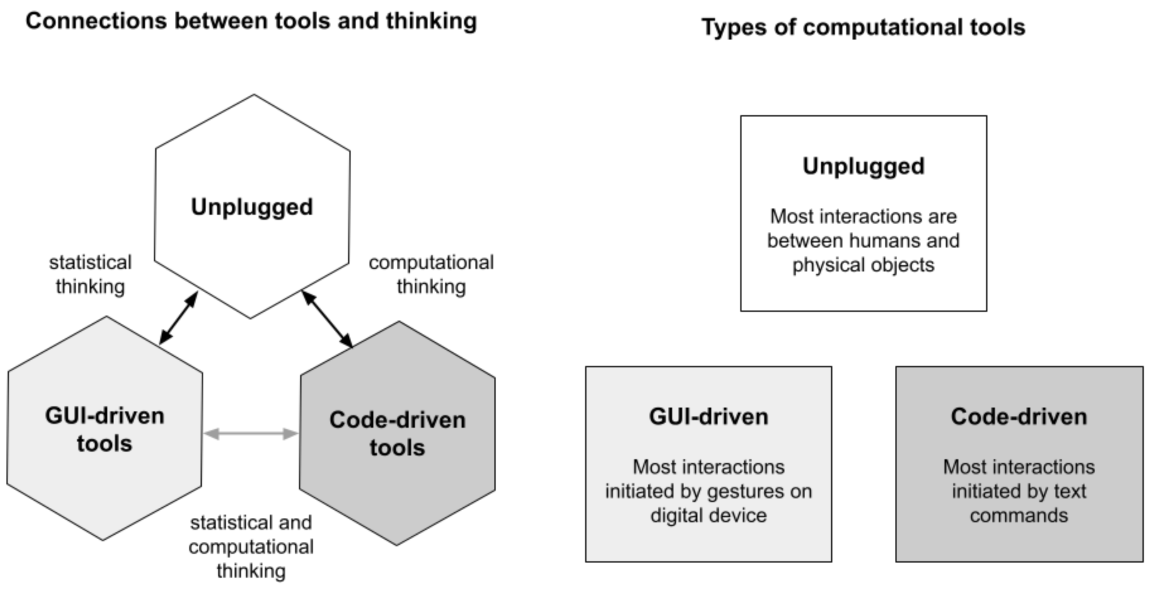

Connections between tools and thinking
Unplugged
statistical thinking
computational thinking
GUI-driven tools
Code-driven tools
statistical and computational thinking
Types of computational tools
Unplugged
Most interactions are between humans and physical objects
GUI-driven
Most interactions initiated by gestures on digital device
Code-driven
Most interactions initiated by text commands

Figure 2.2: Proposed framework to inform the design of tasks involving combinations of different types of interactive learning tools (unplugged, GUI-driven, code-driven)

My framework proposes that all three tools — unplugged, GUI-driven, and code-driven — may be needed to support the development, and integration, of statistical and computational thinking. When selecting and combining different computational tools for teaching statistical modelling, however, considerations are needed about how to support students' mental images and visual representations for each aspect of the modelling process. According to Erickson et al. (2019), the naming of actions that alter a data set's contents or structures as "data moves" might help students move between GUI-driven and code-driven tools. Similarly, I propose that the closer the match between modelling actions carried out with GUI-driven and code-driven tools, the smoother the move for the learner.

Dual coding theory proposes that humans store information in their memory in two distinct forms or representations, verbal and non-verbal, and that memory is enhanced by the use of both representations in learning (e.g., Clark & Paivio, 1991). Using both GUI-driven and code-driven tools allows learners to interact with text-based and graphics-based representations of statistical and computational concepts (e.g., Cook & Goldin-Meadow, 2006). Because learners need to access and integrate new programming concepts with statistical concepts, cognitive load theory (e.g., Sweller et al., 1998) needs to be considered in the design of tasks. Cognitive load theory proposes that humans have limited capacity to their working memory and that the load that tasks impose on working memory should be reduced to optimise performance. To minimise cognitive load, a sequence of simple-to-complex whole tasks needs to be developed (e.g., Wouters et al., 2008).

For students to access and use modern data and associated modelling practices, not only do they need to co-develop statistical and computational thinking, but they also need confidence to use computational tools to support their learning and doing of data science. Students need to develop some awareness and experience with computer programming. Currently, GUI-driven computational tools dominate statistics education research and there is minimal research that has explored the observable thinking practices of learners who engage with statistical modelling using code-driven tools (e.g., Deitrick et al., 2017; Ferreira et al., 2014; Thoma et al., 2018). Given the success of using GUI-driven tools for teaching statistical modelling, and the concepts underpinning it, discarding the use of such tools and replacing them with code-driven tools does not seem a suitable approach for teaching data science at the high school level. Instead, designing tasks that introduce learners to code-driven tools using familiar methods and tools for statistical modelling may be a better route.

2.5 Developing data science tasks that introduce code-driven tools

The literature reviewed in this chapter highlights the need for a careful approach to data science task design that supports learner engagement with statistical concepts at the same time as introducing code-driven tools. Novice data scientists cannot be expected to produce their own interactive learning environments with code, nor expected to use mental imagery or abstraction in the place of visualisations. There is also a need for transparency for the computational actions used, and so an over-reliance on high-level functions within code may not be helpful. Although using GUI-driven tools may support learners to engage with concepts, these tools do not by themselves connect to code-driven tools and these connections need to be part of the design. The interaction between learning tools and tasks, and how the design of both need to change when teaching modelling actions, has implications for teaching statistical and computational thinking.

At the same time as reviewing this literature during 2017 to 2019, I began exploring a blend of unplugged, GUI-driven, and code-driven tools in the statistical modelling tasks that I was developing for my teaching. My exploration and the literature reviewed led to three initial guidelines for developing data science tasks that introduce code-driven tools. The three task design guidelines are:

- Guideline 1: Data science tasks that introduce code-driven tools should be based on structured statistical modelling activities
- Guideline 2: Data science tasks that introduce code-driven tools should explicitly support learners to integrate statistical and computational thinking
- Guideline 3: Data science tasks that introduce code-driven tools should connect unplugged or GUI-driven tool-based modelling actions with code-driven tool-based modelling actions

There is minimal advice for code-driven task design from within statistics education research. As Arnold et al. (2018) noted, task designers, including teachers and researchers, draw on implicit design principles when developing learning activities. Researchers within statistics education and the emerging field of data science education rarely provide rich details of tasks used or discuss the impact of task design features, which is similar to the situation for mathematics education (Watson & Ohtani, 2015). For code-driven tools to have a positive impact on the learning of data science, it is crucial that teachers have access to effective guidelines for task design. Therefore, my research will explore the explication of design principles for the construction of statistical modelling tasks that introduce code-driven tools.

The main and supporting research questions are:

1. What observable thinking practices emerge as teachers, positioned as learners, engage with statistical modelling tasks that introduce code-driven tools?
    - Can these observable thinking practices be characterised as integrating statistical and computational thinking? and if so,
    - What features of the tasks appear to stimulate or support the integration of statistical and computational thinking?
2. What design principles could guide the construction of statistical modelling tasks that introduce code-driven tools?
    - How could tasks be constructed to support the introduction of new sources of data and modelling approaches, simultaneously with new code-driven tools?
    - Does using familiar computational tools or modelling approaches within the same task support the introduction of new code-driven tools?

2.6 Summary

There is a lack of guidelines for making decisions about task design for designing tasks that introduce code-driven tools within the context of statistical modelling. In general, it is difficult to find substantial literature that explicitly communicates strategies for designing tasks that introduce code-driven tools within the context of statistical modelling. This is important because teachers not only need to know how data technologies work themselves, but also how to effectively teach about data technologies. Teaching data science is particularly challenging, because of the number of domains of knowledge learners are expected to combine and co-ordinate. Task design for data science education may also introduce many new knowledge elements in the same task, such as sources of data, approaches to statistical modelling, and computer programming.

As demand for data science education increases, so will the number of teachers involved, adding to the challenge of teacher pedagogical content knowledge (Shulman, 1986). Successful implementation of data science at the senior high school level will be challenging and a strong teacher preparation programme is needed, particularly as teachers at high school may not have advanced degrees in statistics (cf. Bargagliotti & Franklin, 2015) or computer science. Tasks are crucial for learning, and a task design framework is needed to support teachers to be confident and effective designers of tasks for teaching data science at the senior high school level. To produce the task design framework, tasks need to be designed and implemented with teachers, and the features of the tasks and their relationship with observable thinking practices need to be analysed. Chapter 3 describes the design-based research approach used for this study.

Chapter 3: Research methods

3.1 Introduction

Chapter 3 outlines the research approaches, methods used, and the procedures. From an interpretive framework perspective, Section 3.2 presents the paradigm or lens through which my research is conducted. Sections 3.3 to 3.4 outlines the design-based research approach and the problem analysis. Section 3.5 covers the design and implementation of the tasks, including the teaching experiments and participants, data collection and ethics. Sections 3.6 to 3.7 discuss the iterative methods of data analysis, describe the design characteristics, and present considerations about validity and reliability.

3.2 Interpretive framework

All research is interpretive. Researchers make decisions about research procedures based on *their own* beliefs, philosophical assumptions, and theoretical perspectives (Punch & Oancea, 2014) or through an *interpretive framework* or *paradigm* (Denzin & Lincoln, 2011). It is important for researchers to clearly communicate their interpretive framework when describing the nature of any research study, which in my case is *pragmatism* as it aligns with my perspectives on data science education and educational research. Conducting research using a pragmatic interpretive framework supports a focus on developing theories about “what works” and the practical outcomes of the research (Creswell & Poth, 2016; Weaver, 2018). Rather than committing to one system of philosophy or reality, researchers who hold an interpretive framework based on pragmatism can draw on multiple influences, ideas, beliefs, methods, and concepts of knowledge as part of their research process (Creswell & Plano Clark, 2011; Creswell & Poth, 2016). Pragmatists believe that truth is whatever works at the time, and that all research is bound by context, for example, social, historical, political, or cultural contexts (Creswell & Poth, 2016). John Dewey, an influential educator, advanced pragmatism as a philosophy based on human experience, where reflection on beliefs leads to actions, and reflections on actions lead to beliefs (Morgan, 2014b).

I am a teacher as well as a researcher, and so it is natural that my personal teaching experiences have shaped my beliefs about effective teaching and learning. Nearly all my teaching has been at the introductory levels for statistics and data science, rather than at advanced levels involving theoretical probability or mathematics. A significant influence on my beliefs for data science education are the two versions of the New Zealand curriculum (Ministry of Education 1992; 2007), which I taught while at the high school level. The pedagogical practices promoted and embedded in these documents included extensive use of group work, exploration and co-construction of concepts, a de-emphasis on rote learning and memorisation, and teaching of competencies such as relating to others and "learning to learn". My beliefs are also strongly shaped by my participation in statistics education research projects led by Associate Professor Maxine Pfannkuch and Dr Pip Arnold, notably those concerning informal inference (Pfannkuch et al., 2011) and simulation-based inference (Pfannkuch et al., 2013).

As a pragmatist, I believe that knowledge — what is true and real — will change over time and is contextually bound. Within the context of education, knowledge can be viewed as shared social norms on acceptable approaches, and as such must change as the world changes. When designing tasks, I endeavour to create learning experiences that empower all students to be confident, curious, creative, and critical when learning from data. I believe that the use of technology within data science education can be transformative for student learning. Drawing on the pragmatist paradigm, I contend that technology should be used with social learning activities, for example, where students use technology to create products that are shared and discussed with others and where technology is used to support interactions between students and the teacher. Pragmatism, in my opinion, is an important perspective to adopt within the area of data science education research specifically, due to changing nature of data and data-related technologies. What might "work" this year might not work next year, and I believe that research needs to provide practically grounded solutions to teaching and learning issues that make sense for the intended learning context (e.g., high school) and are flexible, transferable, adaptable, and scalable (Hammond & Wellington, 2013; Morgan, 2014a). Because the development of instructional design theory with practical solutions is important for supporting change in teaching practice, a design-based research approach was used to introduce code-driven tools through statistical modelling.

3.3 Design-based research approach

A key feature of design-based research (DBR) is the design and testing of a significant teaching intervention (Anderson & Shattuck, 2012), which for this study was the development of new tasks to *introduce* code-driven tools through statistical modelling. A DBR approach allows for new theories and design principles to be developed that can influence both teaching practice and educational research (McKenney & Reeves, 2018). DBR seeks to develop solutions to practical problems grounded in real learning environments alongside new and reusable design principles (Reeves, 2007). Design principles developed from DBR are theories intended to support other designers to create or observe similar outcomes (Van den Akker, 1999). Design principles are hypothetical in nature and represent both "actionable knowledge and theories of action" (Bakker, 2018, p. 47). They can take the form of domain theories that describe how learners learn or how learning environments influence teaching and learning, or the form of design frameworks that provide a set of design guidelines to use to create a "product" that will support the desired goals for a specific learning context (Edelson, 2002).

The process of DBR is complex and requires ongoing decisions about design that balances the goals and constraints of the research (Edelson, 2002). DBR is flexible and adaptive and as such, there is not one standard DBR process to use. To illustrate the DBR process used for *this research*, I created Figure 3.1 by drawing on the DBR descriptions of Edelson (2002), McKenney and Reeves (2018), and Reeves (2007), and by adapting the DBR process model developed by Hoadley and Campos (2022).

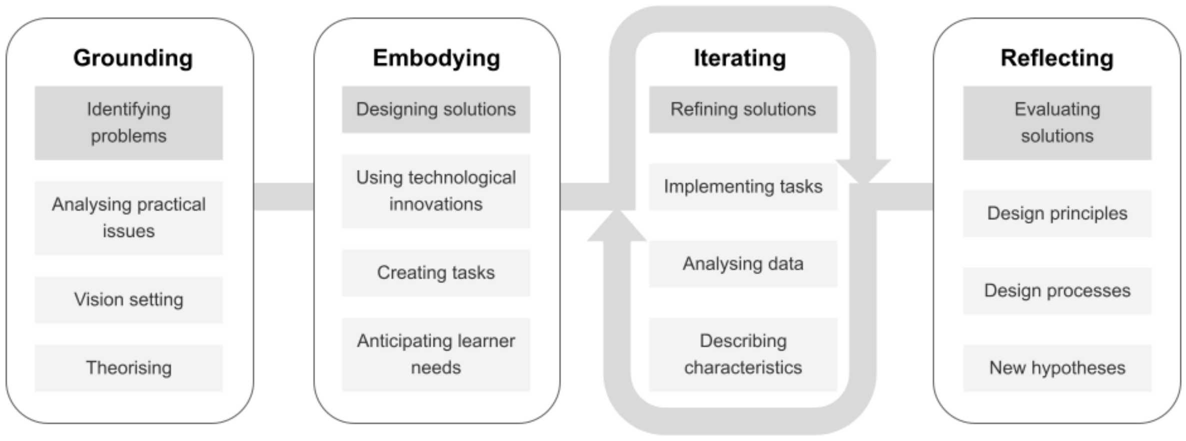


Figure 3.1: DBR process model (adapted from Hoadley and Campos, 2022, p. 212)

As illustrated in Figure 3.1, the DBR process involves interconnected phases of *grounding*, *embodying*, *iterating*, and *reflecting*. In the grounding phase, the designer *identifies complex research problems* by exploring and *analysing practical issues* in collaboration with researchers and teachers. The situated learning context is considered to help *set the vision* for the research and relevant literature is reviewed to *theorise* about potential solutions to the research problem. In the embodying phase, *solutions are designed* that provide learning environments that *use technological innovations*. Drawing on what has been learned in the grounding phase and the designer's knowledge, *tasks are created* by considering the consequences of different design decisions and by *anticipating learner needs*. At the heart of DBR is iteration, where various methods are used across iterative cycles to refine solutions. Methods include *implementing tasks* using teaching experiments, *analysing the data* collected from teaching experiments retrospectively, and *describing the essential characteristics* of the design solution. In the reflecting phase, the designer combines all aspects of the DBR process to *evaluate the solutions* and produce new *design principles*, *design processes*, and *hypotheses*.

Because DBR is not a linear process, I have summarised the research timeline in Figure 3.2 using key milestones.

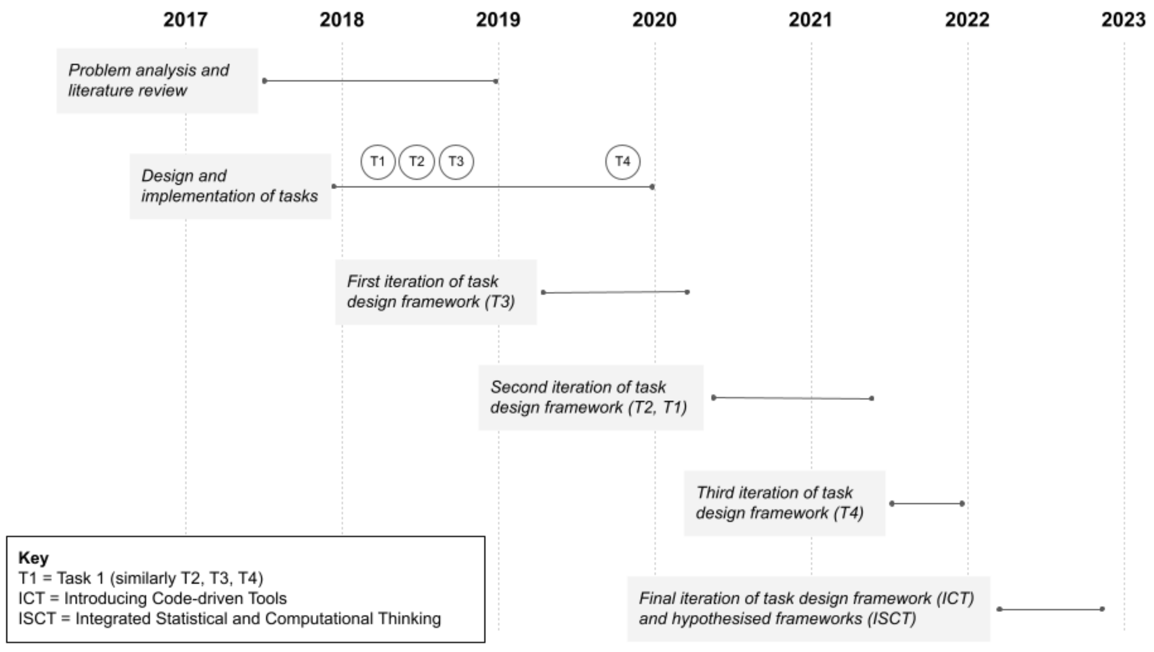

2017
2018
2019
2020
2021
2022
2023
Problem analysis and literature review
T1
T2
T3
T4
Design and implementation of tasks
First iteration of task design framework (T3)
Second iteration of task design framework (T2, T1)
Third iteration of task design framework (T4)
Key
T1 = Task 1 (similarly T2, T3, T4)
ICT = Introducing Code-driven Tools
ISCT = Integrated Statistical and Computational Thinking
Final iteration of task design framework (ICT) and hypothesised frameworks (ISCT)

Figure 3.2: Research timeline

Section 3.4, Section 3.5, and Section 3.6 describe the problem analysis, design and implementation of tasks, and iterative methods used respectively.

3.4 Problem analysis

The research was situated within the wider context of implementing data science at the high school level. Informed by an extensive review of literature and my previous teaching experience, I collaborated with a wider research team (my PhD supervisors) to identify problematic issues with introducing code-driven tools for teaching statistical modelling at the high school level. The problem analysis identified two areas to focus the research on.

The first area of focus considered the practical teaching issue of describing, observing, and teaching an integration of statistical and computational thinking. As discussed in Chapter 2, although frameworks exist for statistical thinking and computational thinking, none appear to explicate their integration with respect to learning from modern data. Hence the first main research question is: *What observable thinking practices emerge as teachers, positioned as learners, engage with statistical modelling tasks that introduce code-driven tools?* Supporting research questions were developed in response to practical issues with designing and implementing tasks: *Can these observable thinking practices be characterised as integrating statistical and computational thinking?* and if so, *What features of the tasks appear to stimulate or support the integration of statistical and computational thinking?*

The second area of focus considered the practical teaching issue of creating learning tasks for teaching data science, specifically statistical modelling. As discussed in Chapter 2, although examples of tasks are shared by data science educators, rarely do the task designers provide explanations or justifications for their task design decisions. Hence, the second main research question is: *What design principles could guide the construction of statistical modelling tasks that introduce code-driven tools?* Supporting research questions were developed in response to practical issues with introducing new data technologies and computational tools: *How could tasks be constructed to support the introduction of new sources of data and modelling approaches simultaneously with new code-driven tools? Does using familiar computational tools or modelling approaches within the same task support the introduction of new code-driven tools?*

The decision was made to carry out research with high school teachers to explore solutions to these problems as this would provide professional development for the teacher participants (van den Akker, 1999), a necessary priority for supporting the implementation of a new technology-based teaching approach.

3.5 Design and implementation of tasks

3.5.1 Task design

Although the focus was the teaching of statistical modelling, the use of code-driven tools required careful consideration of computer science education research and DBR approaches from a technology perspective. The design of tasks for this study did not just involve the development of a sequence of instructions or prompts (task design) but also involved the development of new and innovative technology (tool design). Important task design decisions involved the choice of programming language and the selection of specific code-driven tools in order to provide the coding environments for the tasks. The programming language used for this research was *R* (R Core Team, 2017) and two types of code-driven tools were developed. The first type of code-driven tool involved interactive web pages, which I created using the package *learnr* (Schloerke et al., 2018). This coding environment allowed teachers to execute small "chunks" of *R* code independently within the web browser. The second type of code-driven tool involved RStudio (RStudio Team, 2018), an Integrated Developer Environment. In this coding environment, teachers were given a RMarkdown (Allaire et al., 2018) file that contained both markdown text and code chunks. Both types of code-driven tools made use of the *tidyverse* (Wickham, 2017) ecosystem of *R* packages.

Informed by the initial task design guidelines I developed, and the affordances of the selected code-driven tools and other technologies, four tasks were designed. As there was little research available to guide the design of the tasks, I made frequent use of *thought experiments* to consider what the consequences of different design decisions might be (Bakker, 2018), drawing on my extensive experience with teaching statistical modelling to anticipate learner needs. In line with the DBR process, the tasks were refined iteratively during the implementation process. The details of each of these tasks are discussed in Chapters 4, 5, 6 and 7, and the tasks used are provided in Appendices C, D, E, and F. The tasks were implemented using four different teaching experiments. The details of these teaching experiments are provided in Section 3.5.2. Details about the data collection methods used are provided in Section 3.5.3. Additionally, various versions of the tasks were implemented at non-research workshops with high school statistics teachers and students, and with my introductory level statistics students. These informal implementations of the tasks assisted my design process.

3.5.2 Teaching experiments

Four professional development workshops were developed to trial new educational materials and learning environments for statistical modelling from a data science perspective with high school statistics teachers. All the workshops were conducted in person at the University of Auckland and were facilitated by me. Three workshops were conducted in 2018 and one in 2019. Each of the four workshops contained a teaching experiment.

The first three teaching experiments were conducted in 2018 over three five-hour workshops spaced two weeks apart (26th May, 9th June, 23rd June). The fourth teaching experiment was conducted in 2019 over one five-hour workshop (30th November). A summary of these tasks and teaching experiments is presented in Table 3.1.

Table 3.1: Summary of tasks and teaching experiments used for the research study

Task	Task name	Statistical modelling	School level	Teaching experiment	Workshop date
1	More than fifty shades of grayscale	Classification models	-	1	26th May 2018
2	May the force be with you	Prediction models	Year 13	2	9th June 2018
3	Humans vs Computers	Randomisation test (simulation-based inference)	Year 13	3	23rd June 2018
4	Free the fruity freezes	Probability models (simulation)	Year 12	4	30th November 2019

Across all teaching experiments, teachers worked in pairs with access to one laptop computer, with the pairs decided by me in advance of each workshop. Each task was a learning activity with several distinct phases. Each phase was designed to be completed by each pair independently of the other pairs, and the intent was for me and other members of the research team (my PhD supervisors) to be as “hands off” as possible, offering support and guidance only when requested. All teachers had moderate to strong GUI-based software skills as the current assessment requirements for Year 12 and 13 statistics in New Zealand require the use of software such as spreadsheets, *iNZight* and *VIT*.

3.5.2.1 Recruitment of participants

Advertisements for the study were placed on the Statistics Teachers NZ Facebook page. The Statistics Teachers NZ Facebook page is a publicly moderated group, where teachers share resources and information of interest to teaching statistics. The group had over 500 members at the time the advertisements were placed. I screened potential participants to check that they had experience teaching Year 12 or Year 13 Statistics. Limits were placed on the number of participants for the teaching experiments, due to the need for an even number of participants and teaching space limitations. For the teaching experiments where there were too many volunteers, or an odd number of volunteers, participants were randomly selected. As teachers participated in this research outside of school hours, specifically in the weekends, organisational consent from their respective schools was not needed.

3.5.2.2 Participants for teaching experiments one to three

Participants for the first three teaching experiments were four female and two male Year 13 statistics teachers. On average, the teachers had 10.5 years high school teaching experience (mean = 10.5, min = 7, max = 14) and all but one had at least five years' experience teaching Year 13 statistics (mean = 7, min = 1, max = 12). Only one of the teachers had completed an undergraduate degree majoring in statistics and had any substantive experience with the statistical programming language *R*, although this teacher had not taught *R* at the high school level.

3.5.2.3 Participants for teaching experiment four

Participants for the fourth teaching experiment were eight female and two male Year 12 and 13 statistics teachers. Four of the participants from the first three teaching experiments were also part of the second teaching experiment. On average, the teachers had 12.3 years high school teaching experience (mean = 12.3, min = 4, max = 18) and all but one had at least three years' experience teaching Year 12 or 13 statistics (mean = 10.6, min = 0, max = 18). Four of the teachers had completed an undergraduate degree majoring in statistics. Only one teacher had any substantive experience with the statistical programming language *R*, although this teacher had not taught *R* at the high school level. One of the teachers had completed an undergraduate degree majoring in computer science.

3.5.3 Data collection

Data for the research study was collected from the four teaching experiments. The data collection methods included a questionnaire and observational data captured from teachers completing and discussing the tasks.

3.5.3.1 Questionnaire

A short questionnaire was used to gain information about each teacher (see Appendix A). Teachers were asked how long they had been teaching at the high school level, and specifically how long they had been teaching statistics at the Year 12 or Year 13 level. Teachers were also asked if they had completed a statistics, mathematics, or computer science major as part of their undergraduate degree. Due to the small-scale exploratory nature of this research study, it was not deemed necessary to ask more questions with respect to each teacher's background, as the goal of the analysis was not to generalise findings from a small sample to a bigger population. Instead, the focus for data collection was on capturing each teacher's interactions with each task and the discussions between teachers about the design of the task.

3.5.3.2 Observational data

Teachers worked in pairs with access to one laptop computer and were asked to *think aloud* as they completed each task (cf. *task-based interviews*, Casey & Wasserman, 2015). The *think aloud* method allows for the analysis of thought processes, not just the product of thinking (Van Someren et al., 1994), and provided a way to gain further insight into how the teachers may have been utilising and integrating statistical and computational thinking. Within the task instructions, there were also written prompts asking teachers to discuss certain elements of their approach to the task. To support capturing teachers' thinking and actions while completing the task, reflective practice was encouraged after each task through semi-structured whole-workshop group discussions. During these discussions, teachers were asked to share their beliefs about the viability of the tasks for use with high school students and their personal experiences with completing the tasks.

Teachers also completed a reflection sheet at the end of the third teaching experiment. (See Appendix B for examples of discussion questions and the questions used in the reflection sheet). All the teacher actions, responses, interactions with the software tools and conversations were captured using screen-based video and audio recordings made with browser-based tool *Screencastify*. Additionally, digital documents created during each task were saved, which included artefacts such as RMarkdown files, Google docs, and Google sheets. For some tasks, paper-based handouts were given to teachers to write on and these also provided data. During the teaching experiments I and other members of the research team occasionally offered assistance and guidance to teachers as they were completing the tasks. These interventions were captured in the audio recordings and were included in any transcripts created related to the specific teaching episode.

3.5.4 Ethics

Permission to conduct the study was granted by the University of Auckland Human Participants Ethics Committee for a period of three years from 30th April 2018 (Reference Number 021024). There were several ethical issues that were addressed for this research project. To preserve anonymity in published work, all data were reported in such a way that the participants and their respective schools could not be identified. All personal details of teachers, such as their names, were removed from the transcripts and digital artefacts from the tasks. Pseudonyms were used for each teacher that retained their gender. As the research took place within small professional development workshops where other teachers were present, the researcher included in the participant information sheet and consent form that participants were not to disclose the names of any other teachers who participated in this research to any other person.

Participants were also requested to maintain the confidentiality of information shared by teachers during the workshops. However, participants were also advised in the participant information sheet that it could not be guaranteed that other participants would comply with this request. Written and informed consent was sought and obtained from each teacher before they participated in the research. All participants had the right to withdraw from the study, without giving reason. However, none of the participants withdrew from the study after attending a workshop. Since there may have been a power relationship between myself and each teacher participant, I gave assurance to maintain a professional relationship with each participant. Teachers participated in workshops in their own time during weekends to help resolve the issue of additional work for teachers, and the workshops were spaced at least two weeks apart.

3.6 Iterative methods

In the DBR approach, various methods are used across iterative cycles to refine solutions. Figure 3.3 summarises the iterative process used to design and implement the tasks, retrospectively analyse the data, and describe the essential characteristics of the task design framework.

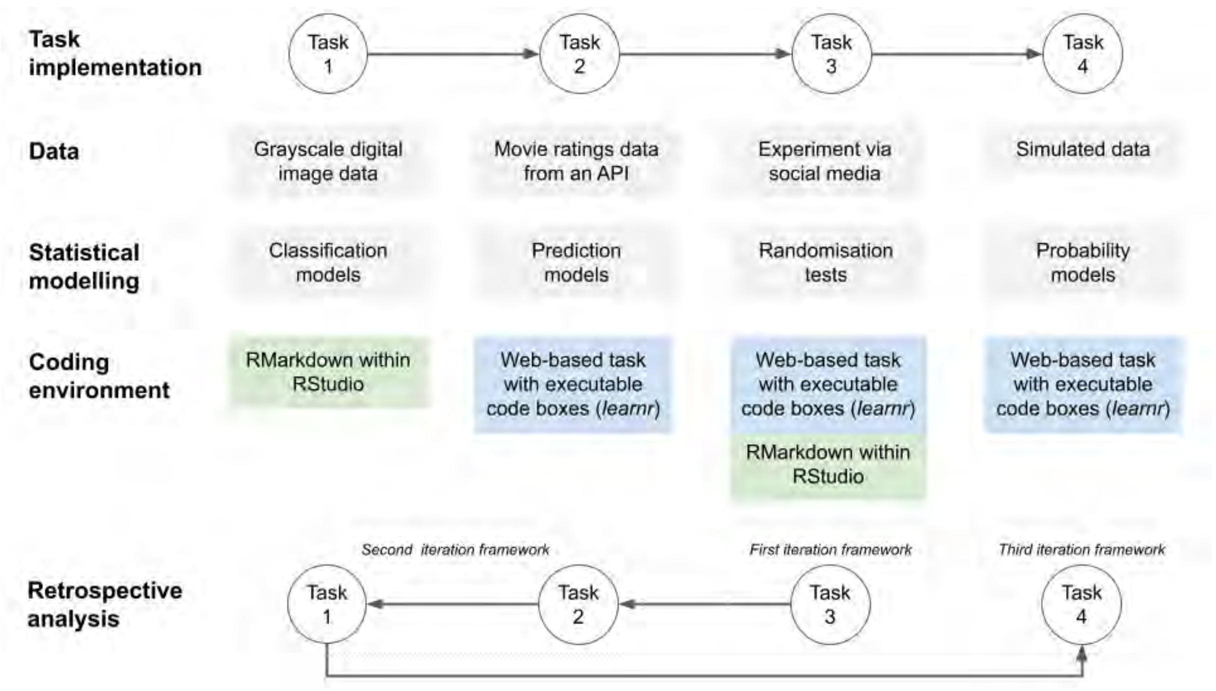

Task implementation
Task 1
Task 2
Task 3
Task 4
Data
Grayscale digital image data
Movie ratings data from an API
Experiment via social media
Simulated data
Statistical modelling
Classification models
Prediction models
Randomisation tests
Probability models
Coding environment
RMarkdown within RStudio
Web-based task with executable code boxes (learnr)
Web-based task with executable code boxes (learnr)
RMarkdown within RStudio
Web-based task with executable code boxes (learnr)
Second iteration framework
First iteration framework
Third iteration framework
Retrospective analysis
Task 1
Task 2
Task 3
Task 4

Figure 3.3: Summary of the iterative methods used in the DBR research

3.6.1 Data analysis

A retrospective and task oriented qualitative analysis approach (Bakker & van Eerde, 2015) was used to explore and find meaning in the teachers' actions, written responses, verbalisations, and conversations. The analysis focused on identifying observable thinking practices and noting what features of tasks appeared to stimulate or support these thinking practices. Starting with the third task (randomisation test), the transcripts and screen recordings were read and viewed, and annotations made to the transcripts with conjectures about the nature of the teachers' thinking and reasoning. Special attention was paid to episodes involving data, models, modelling, computation, and automation, and the computational steps of the statistical modelling activity. These annotations led to the identification of salient examples from each task that would inform the design theory, with examples selected through a process of *constant comparison* (e.g., Bakker & van Eerde, 2015; Creswell, 2012).

The process was then repeated with the second task and first task, before returning to the fourth task. I started with the third task because it was the most traditional of the first three tasks designed and implemented in terms of the statistical modelling approach, and did not involve teachers simultaneously learning about new sources of data and code-driven tools. The results of each iteration of analysis are reported in Chapters 4 to 7. After all four tasks were explored, I compared and contrasted the thinking practices observed across all four tasks in an attempt to characterise integrated statistical and computational thinking. I also considered my expectations and intentions as the task designer for the nature of thinking that could be supported or stimulated by different task features. The results of this analysis were two hypothesised frameworks for Integrated Statistical and Computational Thinking (ISCT), both of which are presented and discussed in Chapter 8.

3.6.2 Description of design characteristics

The data collected from the teaching experiments were used alongside the reflections I made throughout the design process to develop and refine new design principles and processes. These reflections also included any refinements to later tasks that I had made, based on my experience implementing earlier tasks with teachers in the teaching experiments. The retrospective analysis focused on identifying essential characteristics or features of the tasks that promoted or stimulated learning in the domain of statistical modelling. Starting with the third task (randomisation test), I produced a narrative that described my design moves (cf. Hoadley & Campos, 2022) and identified features that I had intentionally included in the task to support and develop statistical and computational thinking.

Where relevant, I linked aspects of my design narrative to literature, to incorporate aspects of research and theoretical perspectives from across statistics, mathematics, and computer science education. Combined with my analysis of the observed thinking practices, I explicated design principles and considerations for constructing statistical modelling tasks that introduce code-driven tools, resulting in the first iteration of a task design framework. This procedure was repeated for the other tasks, using the same order as the data analysis process (see Figure 3.3). A backwards and forwards approach was used to refine and modify the design framework, with changes made to produce reusable design principles, rather than ones that were too specific for each task. The development of the task design framework is presented and discussed in Chapters 4 to 7, and the finalised task design framework is presented in Chapter 8.

3.7 Validity and reliability

For the results of this research to be taken seriously, the research methods used should be valid and reliable. Within a design-based research approach utilising qualitative methods, validity can be evaluated in terms of whether what was measured is the same as what was intended to be measured (Bakker & van Eerde, 2015; Zohrabi, 2013). The findings of a study should be directly related to the problematic issue that motivated the research and the research questions that framed the inquiry; and the methods selected need to be used appropriately and systematically (Burton et al., 2014).

To improve the internal validity, or credibility, of the findings, the data collected should be of high quality and the reasoning used to make conclusions should be sound (Bakker & van Eerde, 2015). For this study, internal validity was enhanced by using tasks that I had piloted previously with students, providing genuine learning activities from the wider context of the study. The teaching experiments took place over an extended period (McKenney & Reeves, 2018), with at least two weeks between each intervention. During the analysis of the data collected from the teaching experiments, conjectures or design principles that were identified from one teaching experiment were checked against previous teaching experiments, and a real effort was made to find counterexamples (Bakker & van Eerde, 2015).

Importantly, data triangulation was used to strengthen the conclusions made by collecting and combining different sources of data during the teaching experiments (Bakker & van Eerde, 2015; Burton et al., 2014; Patton, 2015). For example, the thinking that teachers verbalised using the *think aloud* method was transcribed and compared to the actions observed while completing the task which was captured using a screen recording. During tasks, teachers were also asked to write responses, and all these sources of data were then compared to what teachers said during the group discussions, which were captured using audio recording and then transcribed. The researcher also provided transcripts and screenshots of captured interactions with software tools to other members of the research team to assist with the analysis of the data and to provide additional perspectives (Burton et al., 2014).

With respect to the external validity of the results of a design-based research study, generalisability and transferability need to be considered (Bakker & van Eerde, 2015; McKenney & Reeves, 2018). This was a small exploratory study involving a self-selecting sample of teachers who completed tasks during professional development workshops, which is a very specific context. To enhance the external validity of the design principles, the four tasks developed used four different statistical modelling activities, and the findings are presented to highlight how the task design could be adjusted by others to suit their learners, supported by using thick rich descriptions of salient data (McKenney & Reeves, 2018).

The tasks developed and the design principles were also presented to critical audiences during the research study, including other high school statistics teachers in non-research professional development workshops and other statistics education researchers at conferences (Burton et al., 2014). Audio recordings, transcripts and initial analyses were presented at the International Collaboration for Research on Statistics Reasoning, Thinking and Literacy (SRTL-11 2019, Los Angeles; SRTL-12 2021, Virtual). Additionally, papers based on this research were published in peer-reviewed journals (e.g., Fergusson & Pfannkuch, 2021; Fergusson & Pfannkuch, 2022). The findings of the research study were also compared to relevant literature within the fields of statistics, mathematics, and computer science education to provide stronger theoretical justifications for the proposed design principles.

Reliability with respect to qualitative research methods is concerned with the independence of the researcher (Bakker & van Eerde, 2015), and related issues of dependability, objectivity, confirmability, and transparency (McKenney & Reeves, 2018). Within a quantitative approach, reliability refers to attaining the same results and consistency for measures across different observers or observations (Burton et al., 2014). However, within a qualitative approach reliability is more about whether the data collection and analysis methods are dependable, and that the findings are consistent with these methods and could be repeated (McKenney & Reeves, 2018; Zohrabi, 2013).

To enhance the internal reliability of the research findings, data collection used objective devices such as audio recordings, screen capture recordings, digital artefacts of participant work and written notes or diagrams completed during the teaching experiments. Transcripts were made of salient aspects of the conversations captured, supplemented by screen shots where relevant, and these were shared with other members of the research team during analysis to enable peer examination of the data and enhance the trustworthiness of the research. Teaching experiments conducted in the research study were repeated within non-research settings through professional development workshops to explore if similar responses to the tasks might be obtained.

To improve the external reliability of this study, I took steps to be objective in my analysis by grounding the analysis on the teachers' responses rather than my own motivations for the research (McKenney & Reeves, 2018). However, as analysis does not happen in a vacuum and tends to be driven by the researcher's theoretical interests (Braun & Clarke, 2006), as is the case with design-based research, there was the need for me to actively look for data and responses that contradicted or challenged my pre-conceived views on task design. Furthermore, to demonstrate that the conclusions of the study were based on the design process and teaching experiments, and not just my beliefs, *virtual replicability* was provided by clearly documenting how the research was conducted and how conclusions were made using the data (Bakker, 2018). Specifically, care was taken to explicate the design principles and how these evolved over the research cycles (Sandoval, 2014).

3.8 Summary

In this chapter, I have described the research methods used, and explained how a pragmatic design-based research approach was used to guide the problem analysis, design and implementation of tasks, the iterative methods of data analysis and the description of design characteristics. The next four chapters present the findings for the research and document the development of the Introducing Code-driven Tools (ICT) task design framework. Each chapter focuses on one task and is written in keeping with the four-phase DBR cyclical process model (Figure 3.1) of grounding, embodying, iterating, and reflecting. Each chapter: (1) *grounds* the vision setting in the problem, New Zealand curriculum context, and in the literature that is specific to the task, (2) *embodies* the task through narration of its design; (3) *iterates* through describing the implementation of the task and analysing the data collected; and (4) *reflects* on the DBR process with respect to the Introducing Code-driven Tools (ICT) task design framework and evidence for the integration of statistical and computational thinking.

Chapter 4: Randomisation test task

4.1 Introduction

The implementation of the New Zealand curriculum (Ministry of Education, 2007) had a significant impact on the nature of the statistics taught and assessed in secondary schools. New approaches to statistical modelling, such as simulation-based methods, were introduced into the curriculum and new computational tools were developed to support the teaching and learning of these simulation-based methods (e.g., *VIT Online*, Halstead & Wild, n.d.). Currently, *GUI-driven* tools - computational tools with which users interact predominantly by pointing, clicking, or gesturing within a Graphical User Interface (GUI) - dominate the teaching of statistical modelling at the secondary school level. For secondary school students to be active participants in learning from modern data, however, they need to integrate both statistical and computational thinking, and this necessitates teaching students at least some computer programming skills (Gould, 2010). Integrating computational thinking within the statistics classroom also aligns with the digital technology goals of New Zealand schools.

As computer programming is not currently used within the teaching of statistics in New Zealand secondary schools, it cannot be assumed that statistics teachers have a good knowledge of programming, nor knowledge of how to design statistical modelling tasks that use programming. Therefore, new research is needed that positions teachers as learners, and that explores how to introduce statistics teachers already experienced with GUI-driven tools to *code-driven* tools — computational tools with which users interact predominantly by entering and executing text commands (code). This research aims to build on and extend previous research in GUI-driven statistical modelling (e.g., Garfield et al., 2012) by exploring the design of tasks that facilitate the integration of statistical and computational thinking, as teachers move from GUI-driven tools to code-driven tools within the same learning task.

Using a DBR approach, I explore the design and implementation of a task for introducing teachers to using code-driven tools for randomisation tests (Task 3). Note the rationale for starting the DBR process with the third task is discussed in Chapter 3. I explicate a task design framework to support learners to move from a familiar GUI-driven tool to an unfamiliar code-driven tool and examine what statistical and computational thinking practices emerge when the task is implemented with teachers.

4.2 Teaching simulation-based inference

The last decade has seen growth in the use of simulation-based inference within introductory statistics and data science courses at both the tertiary and secondary school level (Garfield et al., 2012). A simulation-based approach is viewed by many educators as an attractive alternative to a mathematics approach for introducing learners to statistical inference (e.g., Cobb, 2007). Technology has enabled the development of more intuitive tools for supporting and visualising simulation-based inference than what was available when computers were first used in classrooms (e.g., Ben-Zvi, 2000). There is also no need to master algebraic representations before engaging in formal statistical inference (Forbes et al., 2014) so mathematics is not a barrier to learning.

The use of simulation also promotes a modelling perspective for learning (Garfield et al., 2012) and gives learners the opportunity to construct their own knowledge of statistics (Gelman & Nolan, 2017). I define statistical modelling as the process of learning from data where one constructs or selects and then uses a model for the underlying system that produced that data and where one is aware of, and attempts to account for, uncertainty associated with the data and/or model. There are two broad uses of statistical models to learn from real observed data: (1) to generate individual outcomes or predictions, (2) to summarise underlying distributions or processes, including testing hypotheses. As modelling processes may involve both data collected directly from a real process and data generated via simulation from a model, there is a need to define data as "real" or "model-generated" more clearly, in line with several proposed statistical modelling frameworks (e.g., Case & Jacobbe, 2018; Fergusson, 2017). There is also a need to provide transparency of process, and with respect to using "unplugged" activities, simulations allow for learners to use physical actions to represent steps in the modelling process (Wood, 2005).

The amount, availability, diversity, and complexity of data that is now available in the modern world requires educators to broaden their definitions of what data is and what it means to learn from data (Finzer, 2013; Gould, 2010). Statistical modelling with modern data requires knowledge that extends beyond statistical, to include the computational. A focus on developing computational thinking will also help learners navigate an increasingly digital society (Bell & Roberts, 2016). Definitions of computational thinking vary, from broad statements such as "computational thinking is the thought processes involved in modelling a situation and specifying the ways an information-processing agent can effectively operate within it to reach an externally specified (set of) goal(s)" (Nardelli, 2019, p. 34), to multi-dimensional frameworks (e.g., Brennan & Resnick, 2012) similar to the four-dimensional framework (investigative cycle, interrogative cycle, fundamental types of thinking, dispositions) developed for statistical thinking by Wild and Pfannkuch (1999).

At the undergraduate level, programming is presented as an essential computing skill by those advocating for modernising the statistics curriculum (e.g., Cetinkaya-Rundel & Rundel, 2018; Nolan & Temple Lang, 2010) and is an important enabler of developing "data acumen" (NASEM, 2018). Computer programming does provide an efficient way to test learners' computational thinking (Bell & Roberts, 2016), but computational thinking is not the same as programming (Wing, 2006) and can be developed without using code-driven tools. For example, employing computer science unplugged activities, Bell et al. (2009) had learners working in teams using a problem-solving approach that achieved a goal without using a computer.

Although there are existing frameworks related to statistical thinking and modelling (e.g., Grolemund & Wickham, 2014; Wild & Pfannkuch, 1999), there appear to be none that specifically characterise the *integration of statistical and computational thinking* (c.f. Weintrop et al., 2016). Key elements of *statistical thinking* include reasoning with statistical models and integrating statistical and contextual knowledge (see Wild & Pfannkuch, 1999). With respect to statistical learning task design, Weiland's (2016) framework proposes that the highest level is a task where the context is a critical component in making sense of the task and must be considered to answer the task. A research challenge to be resolved is how to integrate statistical and computational thinking when learning from modern data, and therefore how to design tasks for such a teaching approach.

While Cobb's (2007, p. 12) call to place the permutation or randomisation test as the "central paradigm for inference" did not specify the use of particular software tools, the software tools developed or used by statistics education researchers have been predominantly GUI-driven (e.g., Garfield et al., 2012). Statistics education researchers have considered carefully how GUI-driven tools influence learners' thinking when learning statistics and have also considered how to define the complex relationship between software features, task design and learners' statistical conceptions (e.g., Ben-Zvi, 2000; Fergusson, 2017). Statistics educators have also considered the relationship between tool, task and thinking for modelling activities (e.g., Biehler, 2018).

The popularisation of the field of data science has led statistics education researchers to re-think their ideas about tools for teaching statistical modelling. Kaplan (2007) presented a case for using code from the programming language *R* (R Core Team, 2017) to teach the randomisation test, arguing that a barrier to teaching computational-based methods is not computing but the language or notation used to describe things. It has only been in recent years, however, that statistical computing educators have developed packages (collections of *R* code functions) specifically designed to support novices to learn simulation-based inference through the careful design of functions and consideration of syntax. Some examples of *R* packages developed include the *mosaic* package (Pruim et al., 2017), the *infer* package (Bray et al., 2018) and the *mobilizr* package (Molyneux et al., 2017).

Recently there have been discussions among statistics educators about the design of learning activities that may better prepare learners for using code-driven tools (e.g., Biehler, 2018; McNamara, 2015). There is limited research at the secondary school level, however, involving code-driven tools (e.g., Ferreira et al., 2014) and there appears to be no research in which learners engage with statistical modelling tasks specifically designed to move them from using a GUI-driven tool to a code-driven tool within the same learning task. Using any computational tool for teaching statistical modelling requires pedagogical considerations. For example, how much should the instructor and/or software hide, and how much should it reveal? As "an instructor may use graphical interfaces and high-level functions to hide many details of computations from students" (Hesterberg, 1998, p. 7), *computational transparency* is a real pedagogical issue to consider, particularly within the context of computational simulation and notions of glass box versus black box simulations (Magana et al., 2011). In relation to the use of GUI-driven and code-driven tools for statistical modelling, I define *computational transparency* as the degree to which the computations performed by the tool are obvious to the user (c.f. realtime, n.d.).

Before learners can ask, (1) What do I want the computer to do for me? and (2) What does it need to know in order to do that? (Pruim et al., 2017), learners need to understand the statistical problem they want the computer to help them solve. While code-driven tools are a natural alignment to the computational process of simulation-based inference, current GUI-driven tools may offer greater access and support for statistical concepts. There are also considerations to be made about how to support learners' mental images and visual representations of each aspect of a statistical modelling process, aspects that may not be well translated to a code-driven tool through current programming pedagogy. The motivation for introducing code-driven tools is not to replace GUI-driven tools for statistical modelling. Instead, I see the introduction of code-driven tools as an *expansion* of the learner's toolkit for learning from modern data. I propose that the integration of statistical and computational thinking could be supported by building on learners' experiences and knowledge of a GUI-driven tool for statistical modelling and scaffolding their learning towards a code-driven tool.

4.2.1 Theoretical frameworks

In an attempt to bring together learning tools from across statistics and computer science education I developed a framework based on the literature to inform the design of tasks involving unplugged, GUI-driven and code-driven learning tools and the definition of these tools (Figure 4.1).

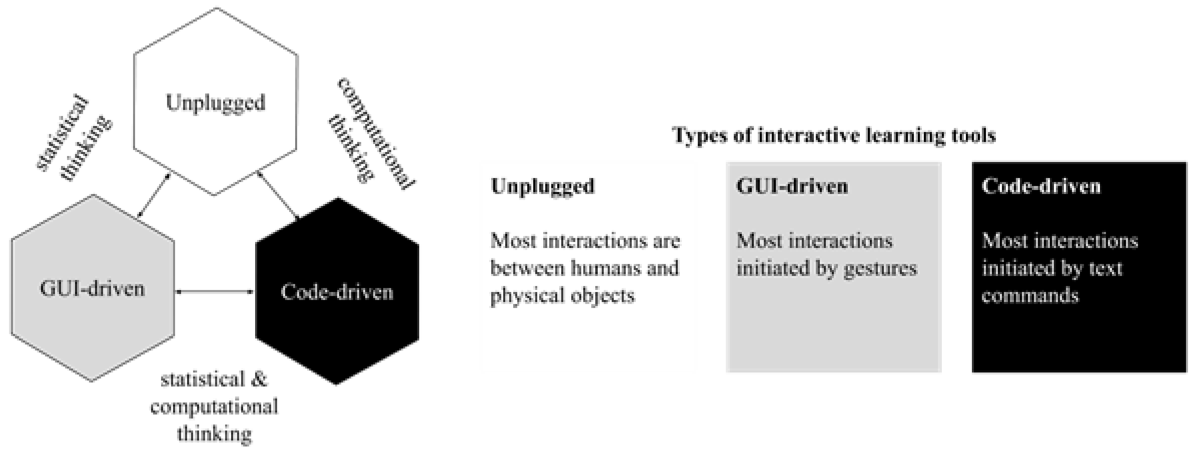

Unplugged
statistical thinking
computational thinking
GUI-driven
Code-driven
statistical & computational thinking
Types of interactive learning tools
Unplugged
Most interactions are between humans and physical objects
GUI-driven
Most interactions initiated by gestures
Code-driven
Most interactions initiated by text commands

Figure 4.1: Proposed framework to inform the design of tasks involving combinations of different types of interactive learning tools (unplugged, GUI-driven, code-driven)

Furthermore, a research gap was identified regarding principles for task design to ensure learners move smoothly between GUI-driven tools and code-driven tools. According to Erickson et al. (2019) the naming of actions that alter a data set's contents or structures as "data moves" might help students' move between tools such as CODAP (GUI-driven) and *R* (code-driven). Similarly, I conjectured that the closer the match between modelling actions carried out with GUI-driven and code-driven tools, the smoother the move for the learner.

Supporting the use of the proposed framework to develop tasks is the dual coding learning theory (e.g., Clark & Paivio, 1991), as the blending of GUI-driven and code-driven tools will expose learners to text-based and graphics-based representations of statistical and computational concepts through different types of interactions with the tools (e.g., Cook & Goldin-Meadow, 2006). Dual coding theory proposes that humans store information in their memory in two distinct forms or representations, verbal and non-verbal, and that memory is enhanced by the use of both representations in learning (e.g., Clark & Paivio, 1991). As learners need to access and integrate new programming concepts with statistical concepts, cognitive load theory (e.g., Sweller et al., 1998) needs to be considered in the design of tasks.

Cognitive load theory proposes that humans have limited capacity to their working memory and that the load that tasks impose on working memory should be reduced to optimise performance. To minimise cognitive load, a sequence of simple-to-complex whole tasks needs to be developed (e.g., Wouters et al., 2008). Thus, task development requires consideration of types of tools, movement between tools in a statistical modelling activity, dual coding learning and cognitive load theories, and existing statistical modelling tasks (e.g., Garfield et al., 2012). The following computer science teaching strategies identified by Sentance and Csizmadia (2017) should also be considered: unplugged type activities, contextualising activities, collaborative learning, developing computational thinking and scaffolding programming tasks.

4.3 Task design framework

Even though the task design framework was developed after the implementation, it is now presented to show how it relates to the features of the task that are discussed in Section 4.4. Through a retrospective analysis of the task, I explicated the *first iteration* of a design framework to construct tasks that introduce learners to code-driven tools through statistical modelling. The analysis involved the research team conducting an intense and robust interrogation of me, as the designer of the task, and the task itself, to identify, name, and collaboratively define the design principles and associated considerations. The six design principles of the framework are described below.

Principle 1: Immerse in data context. Learners participate in activities that require minimal statistical or computational knowledge. The focus is on engaging learners with the data context, understanding the nature of the data to be used for the task and on stimulating interest.

Principle 2: Re-familiarise with GUI-driven tool. Learners use a GUI-driven tool with which they already familiar to carry out a statistical modelling activity. The focus is on statistical thinking.

Principle 3: Describe computational steps of GUI-driven tool. Learners are asked to describe and write down the key computational steps of the GUI-driven tool in their own words. The focus is on encouraging learners to think computationally.

Principle 4: Match computational steps with corresponding code chunks. Learners are presented with individual code chunks that each match the computational steps of the GUI-driven tool. The focus is on reading and attempting to make sense of the code presented.

Principle 5: Re-use code chunks with slight modifications. Learners re-use the same code chunks to create the same statistical modelling activity with different variables or a different data set within the same data context. The focus is on encouraging an integration of statistical and computational thinking by making small changes to the code chunks in response to the change in data, to produce an expected output.

Principle 6: Explore "what if?" changes to code. Learners are encouraged to be creative and to explore "What if?" scenarios. The focus is on modifying at least one aspect of the provided code to produce new or unexpected outputs, and so further develop the integration of statistical and computational thinking.

Three design considerations are also proposed when creating tasks using these design principles.

Consideration 1: The data used. Different data sets used should be within the same data context and should have different features that can be exploited to stimulate both statistical and computational thinking.

Consideration 2: The introduction of new knowledge. Each phase of the task should focus on the use of one tool and should introduce at most one new knowledge. New knowledges may be statistical, computational, context-related, data-related or tool-related.

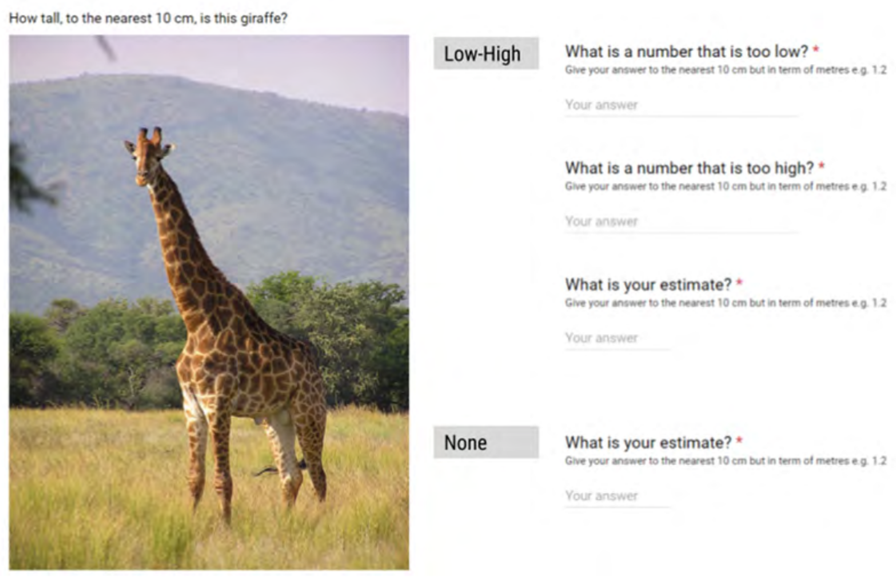
Consideration 3: The level of computational transparency. Under the design principles, identify the key computational steps of the GUI-driven tool for the statistical modelling task, and then develop code chunks to match each computational step. As each computational step identified could be represented in many different ways using code, decide the level of computational transparency that should be used across the learning task.

4.4 Task design characteristics

The statistical modelling focus for this task was a simulation-based approach to inference, specifically the randomisation test. The statistical modelling process involved using a null model, simulating data from the null model through re-randomisation, and using the re-randomisation distribution of the test statistic to make a statistical inference (Garfield et al., 2012). It was conjectured that the task would simulate thinking following a path of statistical (Phases 1 & 2), computational (Phases 3 & 4), to an integration of the statistical and computational (Phases 5 & 6). Each phase of the task (see also Appendix C) is now described with respect to the proposed design framework presented in Section 4.3.

Phase 1: Immersing the teachers in the data context for the task.

The purpose of this phase was to ground teachers in a rich data context for statistical modelling. The data context selected was humans' ability to estimate heights and whether the height estimates could be influenced. An initial activity involved the teachers estimating the heights of trucks and predicting whether a particular truck would fit under a bridge. A news clip was then shown describing the use of sensors to detect trucks that might be over the height limit for an upcoming bridge. I then described to the teachers how I developed an experiment involving estimating the height of a giraffe in meters. Two different versions of the Google form were created. Both forms featured the same picture of a giraffe, but one version of the form used two "prompting" questions and the other version used no prompts (Figure 4.2).


How tall, to the nearest 10 cm, is this giraffe?

Low-High

What is a number that is too low? *
Give your answer to the nearest 10 cm but in term of metres e.g. 1.2

Your answer

What is a number that is too high? *
Give your answer to the nearest 10 cm but in term of metres e.g. 1.2

Your answer

What is your estimate? *
Give your answer to the nearest 10 cm but in term of metres e.g. 1.2

Your answer

None

What is your estimate? *
Give your answer to the nearest 10 cm but in term of metres e.g. 1.2

Your answer

Figure 4.2: The two versions of the Google form: Low-High and None

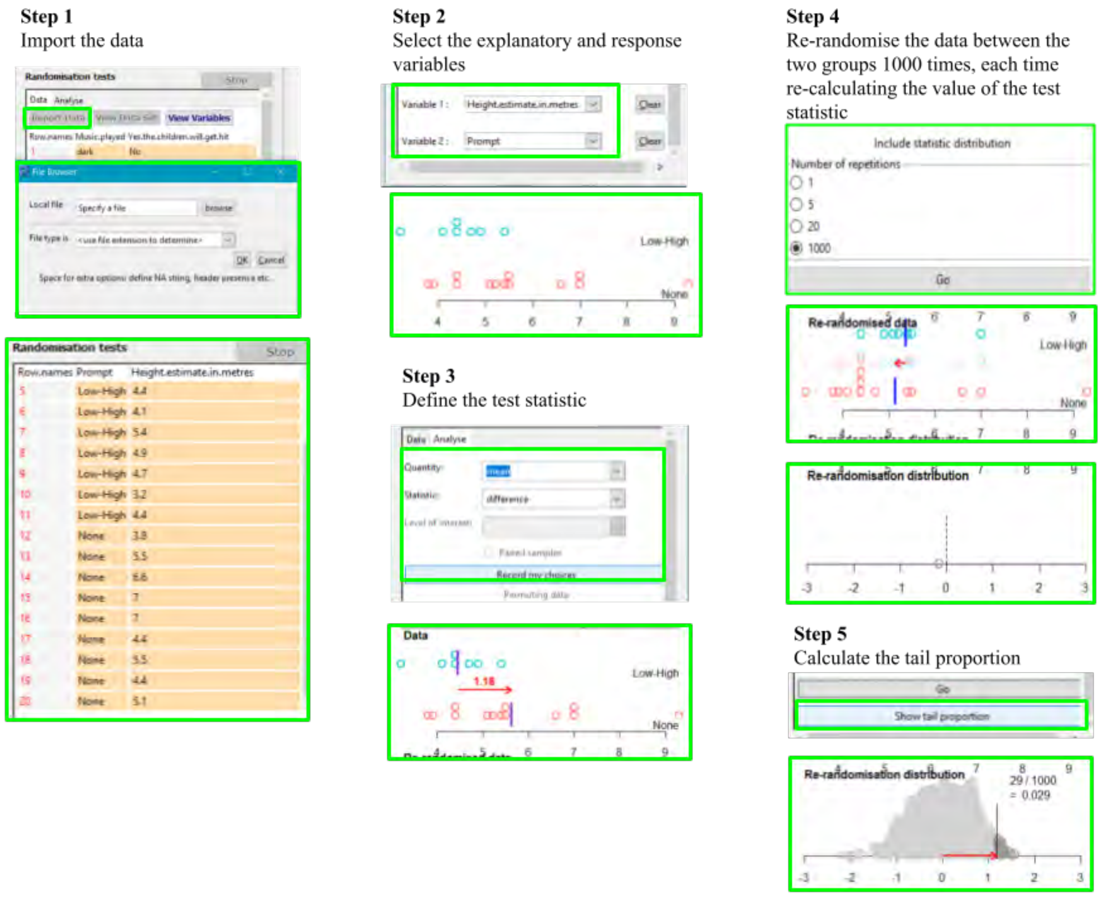
I then explained how I shared a link on Twitter and Facebook asking for volunteers to complete the survey, and how this link randomly allocated a person to one of the two versions of the survey using code. The modern context of conducting experiments online through social media platforms allowed for a data science perspective on an otherwise traditional experimental design.

Phase 2: Re-familiarising teachers with using the GUI-driven tool VIT Online for carrying out the randomisation test.

The purpose of this phase was to familiarise teachers with the data from the giraffe experiment discussed in Phase 1, including the names of the variables and how data from each variable was recorded, and to stimulate statistical thinking by considering a causal claim. Teachers were asked to explore what could be learned from the data by carrying out a randomisation test using the GUI-driven tool *VIT Online* (Wild & Halstead, n.d.), a tool that they were familiar with. A key design feature of *VIT Online* is that each user interaction with the tool produces visual feedback, including graphics and animations. For the randomisation test, users import the data into *VIT Online*, select the response and explanatory variables and then decide what statistical measure will be used to compare the groups (the test statistic). The user then selects an option in the tool to re-randomise the group labels to the response variable 1000 times. For each re-randomisation, the test statistic is calculated and the re-randomised (or simulated) test statistics are collected in a distribution called the re-randomisation distribution. The test statistic is compared to the re-randomisation distribution and the tail proportion is calculated by determining how many of the 1000 simulated test statistics are at least as large as the observed test statistic. At the end of this phase, each pair of teachers shared what they had learned.

Phase 3: Describing the computational steps of the GUI-driven tool VIT Online for carrying out the randomisation test.

The purpose of this phase was to focus teachers' attention on general aspects of the computational sequence, in particular the flow of information between the user and the tool. Teachers were given an A3 sheet of paper with five *VIT Online* screenshots for the randomisation test using the giraffe data. Each screenshot represented one computational step and green "highlight" boxes were used in the screenshots to focus teachers' attention. Only the step numbers were provided to teachers, not the description of the step (see Figure 4.3).



Randomisation tests

Row.names	Prompt	Height.estimate.in.metres
5	Low-High	4.4
6	Low-High	4.1
7	Low-High	5.4
8	Low-High	4.9
9	Low-High	4.7
10	Low-High	3.2
11	Low-High	4.4
12	None	3.8
13	None	5.5
14	None	6.6
15	None	7
16	None	7
17	None	4.4
18	None	5.5
19	None	4.4
20	None	5.1

Figure 4.3: Key computational steps on A3 sheet for the randomisation test with the giraffe data when using VIT Online

Teachers were given the instruction to discuss and write down for each step what the computer had to do "behind the scenes" to interpret or use what they had done (as the user) to produce the output or changes in the GUI. The expectation was not that teachers could describe the specifics of the code used but that they could decompose and explain the computational aspect of the modelling process in their own words.

Phase 4: Matching the computational steps with code chunks.

As the teachers already knew what each step of code produced, the purpose of this phase was to encourage teachers to link aspects of their descriptions of each computational step to the "words" used in the code. Teachers were progressively given chunks of code that matched each of the five computational steps for the randomisation test presented in Phase 3. Intentionally, the code provided did not produce graphics for Steps 2 to 5. The programming language *R* (R Core Team, 2017) was used, and the code-driven tool was an interactive web page created using the package *learnr* (Schloerke et al., 2018). This tool allowed teachers to execute small "chunks" of *R* code within a web browser. The *tidyverse* (Wickham, 2017) ecosystem of *R* packages was used for the code.

The *infer* package (Bray et al., 2018) was used for functions relating to conducting a randomisation test, as the functions provided by the package were the closest match to the computational steps for the randomisation test using *VIT Online* and gave some transparency to the computations being performed. I developed additional functions and used these in the code provided to teachers. Teachers were instructed that for each step on the A3 sheet, they needed to run the code chunk supplied, discuss what they thought the code meant and add a comment in the first line to summarise what the code did for this step. Some comments were already provided within the code and these have been indicated for the reader in Figure 4.7 by adding "(provided)" to the end of the comment line.

Phase 5: Re-using code chunks to carry out the randomisation test.

The purpose of this phase was to stimulate an integration of statistical and computational knowledge, by requiring teachers to re-use the supplied code with small changes induced by properties of the new data set. Teachers were introduced to a similar experiment involving my stage one statistics students estimating the height of a man from a picture, and were asked what could be learned from these data with respect to whether the use of prompts influenced height estimates. The stimulus for this new experiment was a *man* not a *giraffe*, which provided new contextual information that the teachers could reason with. Unlike Phase 4, where teachers had used the same giraffe data from Phases 2 and 3, the *man* data was completely unfamiliar to them and had different statistical properties. The variable names were the same but the levels of the explanatory variable (Prompt) changed from "High-Low" and "None" to "yes" and "no".

Additionally, the height estimates of the man for the two groups were very similar, both in terms of central tendency and spread.

Phase 6: Exploring "What if?" changes to the code by creating "new" test statistics.

The purpose of this phase was to develop further an integration of statistical and computational thinking by encouraging teachers to explore "new" test statistics beyond the difference of two means or medians. Teachers were given a RMarkdown (Allaire et al., 2018) file and used this within the code-driven tool `RStudio` (RStudio Team, 2018). The RMarkdown file contained text instructions and all the code chunks from Phases 4 and 5, and was "knitted" (rendered) by the teachers to produce HTML output. The code used for all steps in this phase was identical to the code used in Phases 4 and 5 but also included the addition of new functions that produced graphics similar to those produced by *VIT Online.* The code provided to teachers used the giraffe data but teachers were able to change the data to the man data by changing the code. Teachers were asked to create a new test statistic based on what they had noticed was different between the two groups.

4.5 Analysis

The implementation of the randomisation test task took place during the third day of the professional development workshops. I focus on two of the teacher participants, Naomi and Ingrid, to provide a rich description of their actions and reasoning. During Phases 1 to 5, Naomi and Ingrid's actions and reasoning were similar to the other two pairs of teachers. During Phase 6, Naomi and Ingrid responded to the instruction to explore new test statistics, whereas the other two pairs of teachers explored different "What if?" scenarios.

Both Naomi and Ingrid had taught at state-funded secondary schools for at least four years, with most of these years at single-sex girls' schools. Naomi had 12 years' experience teaching Year 13 statistics, but had not completed an undergraduate degree majoring in statistics and did not have experience with the statistical programming language *R*. Ingrid was in her first year of teaching Year 13 statistics, had completed an undergraduate degree majoring in statistics and had experience with the statistical programming language *R*.

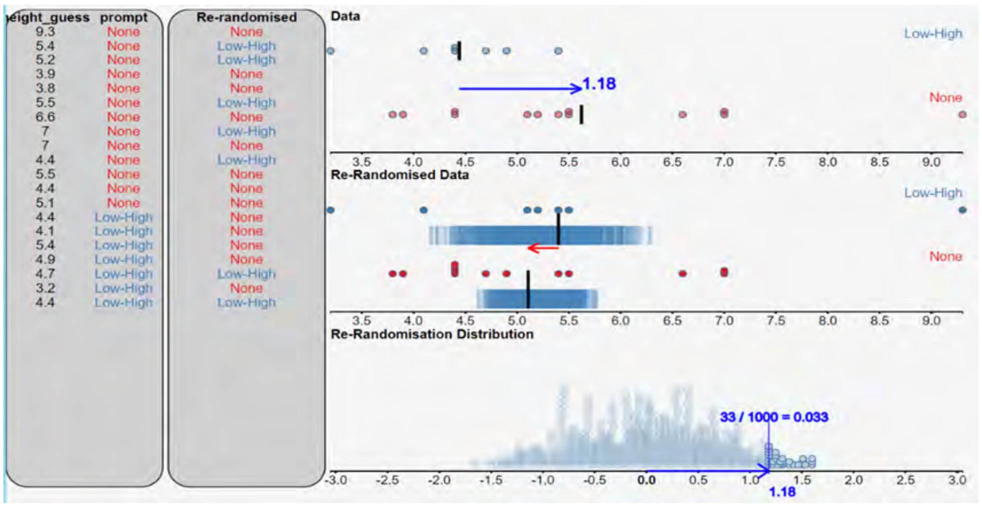
The results are presented sequentially for each phase of the task. Naomi and Ingrid spent 15 mins, 10 mins, 25 mins, 16 mins, 12 mins, and 14 mins working within each of the six phases of the task respectively. Reproductions have been made of the teachers' writing and interactions with the statistical software tools for graphics quality purposes, based on the screen recordings made during the workshop.

4.5.1 Phase one: Immersing the teachers in the data context for the task

The conjectured thinking for this phase was statistical, with a focus on developing contextual knowledge to inform the later phases of the task. The teachers shared their estimates of heights and participated in discussions about how the use of context and external cues may influence estimation processes. For example, when given a photo of a bridge, Naomi and Ingrid estimated the height under the bridge to be five meters but said that it was difficult to tell. However, when given a photo of a truck alongside the photo of the bridge and asked whether they thought the truck would make it under the bridge, Naomi responded "that's easier now because you're providing more scale." After learning about the design for the experiment and the variables used, Naomi and Ingrid expressed excitement at "finding out what happened" with the *giraffe* experiment.

4.5.2 Phase two: Re-familiarising teachers with using the GUI-driven tool VIT Online for carrying out the randomisation test (10 mins)

The conjectured thinking for this phase was statistical, with a focus on using a statistical model and integrating statistical and contextual knowledge. As Naomi and Ingrid were already familiar with the concept of a randomisation test and with implementing such a test using a GUI-driven tool, the teachers successfully used *VIT Online* to carry out a randomisation test (Figure 4.4).



eight_guess	prompt	Re-randomised
9.3	None	None
5.4	None	Low-High
5.2	None	Low-High
3.9	None	None
3.8	None	None
5.5	None	Low-High
6.6	None	None
7	None	Low-High
7	None	None
4.4	None	Low-High
5.5	None	None
4.4	None	None
5.1	None	None
4.4	Low-High	None
4.1	Low-High	None
5.4	Low-High	None
4.9	Low-High	None
4.7	Low-High	Low-High
3.2	Low-High	None
4.4	Low-High	Low-High

Figure 4.4: Randomisation test for difference of two means using giraffe data

Naomi demonstrated an understanding of statistical modelling when she stated, in reference to the re-randomisation distribution, that "you're always going to get something symmetrical around zero and the question is, how much, how big is this [referring to the observed difference] compared to zero?" Although the statement about comparing the observed difference to zero is partially correct, the statement about the general features of a re-randomisation distribution is incorrect. While in this learning context it may seem reasonable to assume that Naomi was only considering a randomisation test for a difference of two means, her comments in the later phases of this task revealed that she did in fact believe that all re-randomisation distributions were symmetric and centred around zero. Naomi then used the randomisation test output to make a causal claim saying, "OK, so we're fairly certain that people who get no prompts tend to make higher estimates than people who are given a low prompt first." Although this statement does demonstrate statistical thinking, it should be noted that the causal claim made does not refer to the design of the study.

The teachers' discussion also extended beyond the test for the difference of two means into claims of causality for variation:

> Naomi: There's also more consistency. Doing the Low-High thing brings more consistency.
>
> Ingrid: Yeah, the spread variation is quite important.

The integration of statistical and contextual knowledge, which is an element of statistical thinking, was demonstrated during their discussions with the other teachers in the study, by linking the variability in height guesses to humans being uncertain. Naomi and Ingrid proposed that variability was an aspect of the data distributions that was overlooked by students when doing randomisation tests and that the variability was the more interesting aspect to the data than the centres.

4.5.3 Phase three: Describing the computational steps of the GUI-driven tool VIT Online for carrying out the randomisation test

The conjectured thinking for this phase was computational. As expected, the teachers' discussions during this phase were mostly about what was being calculated, how data were processed or generated, and what visualisations were provided, for each of the five steps of the randomisation test using *VIT Online*. For all the steps in this phase, Ingrid and Naomi wrote notes that separated the visual graphics produced by the tool from the computations performed to carry out the randomisation test and discussed the difference between what the "computer" needed to perform a calculation and what the "human" needed to see.

Naomi and Ingrid's computational thinking for Step 4 (see Figure 4.3) did require some additional statistical knowledge about simulating data from the null model. After looking at the screenshot provided for Step 4, Ingrid began to describe her understanding of what was being done by the computer.

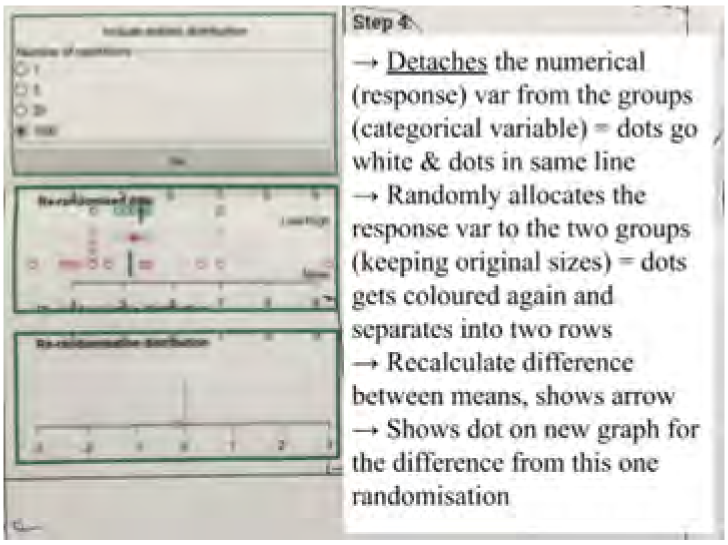
> Ingrid: For me, it's putting all the data into one distribution and uses that distribution as the null distribution, as the distribution of "what if it [the treatment] doesn't matter?"
>
> Naomi: It [the computer] doesn't need to do all of that, all it does, it detaches the values from the groups and randomises the numeric data. I would think it was taking a random sample, but I don't actually know how the computer does it.

The discussion is not yet about calculating the test statistic from the re-randomised data, but about the re-randomisation process. At this point, the teachers went back to *VIT Online* tool which still had the giraffe data loaded and watched more closely as the tool visualised one re-randomisation.

> Naomi: So, see it's detaching the groups, it deallocates it and it separates it out again. You could think about it as randomly separating into two groups.

Ingrid: So it randomly re-allocates the observations, the responses.

After re-examining the visualisations provided from *VIT Online*, the teachers settled on a description of the re-randomisation step as "randomly allocates the response variable to the two groups (keeping original sizes)" (Figure 4.5).



Step 4

→ Detaches the numerical (response) var from the groups (categorical variable) = dots go white & dots in same line

→ Randomly allocates the response var to the two groups (keeping original sizes) = dots gets coloured again and separates into two rows

→ Recalculate difference between means, shows arrow

→ Shows dot on new graph for the difference from this one randomisation

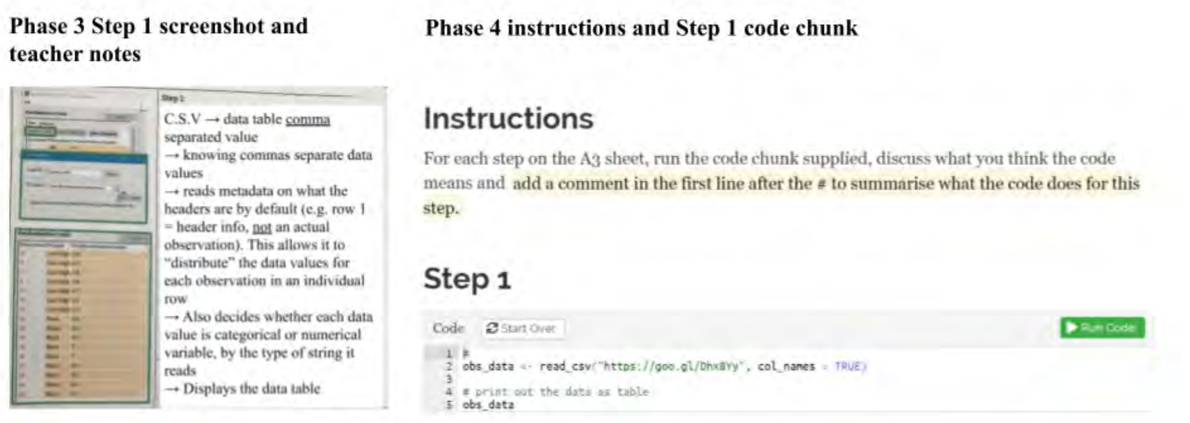
Figure 4.5: Step 4 provided VIT screenshot with teacher notes

It appeared that the instruction to describe the computational nature of this step also helped the teachers to strengthen their statistical understanding of the re-randomisation process.

4.5.4 Phase four: Matching the computational steps with code chunks

The conjectured thinking for this phase was computational, and it was expected that teachers would be able to link each GUI-driven tool interaction with functions used in code. Across all the steps, the teachers demonstrated making connections between the computational aspects of the GUI-driven and code-driven tools.

For example, for Step 1 Naomi and Ingrid immediately recognised that the code matched an aspect of what they had discussed for Step 1 in Phase 3, with Ingrid stating, "the `col_names = TRUE` is what we were talking about with that first row, saying yes do read the first row as col names" (see Figure 4.6).


Phase 3 Step 1 screenshot and teacher notes

Step 1:

C.S.V → data table comma separated value
→ knowing commas separate data values
→ reads metadata on what the headers are by default (e.g. row 1 = header info, not an actual observation). This allows it to "distribute" the data values for each observation in an individual row
→ Also decides whether each data value is categorical or numerical variable, by the type of string it reads
→ Displays the data table

Phase 4 instructions and Step 1 code chunk

Instructions

For each step on the A3 sheet, run the code chunk supplied, discuss what you think the code means and add a comment in the first line after the # to summarise what the code does for this step.

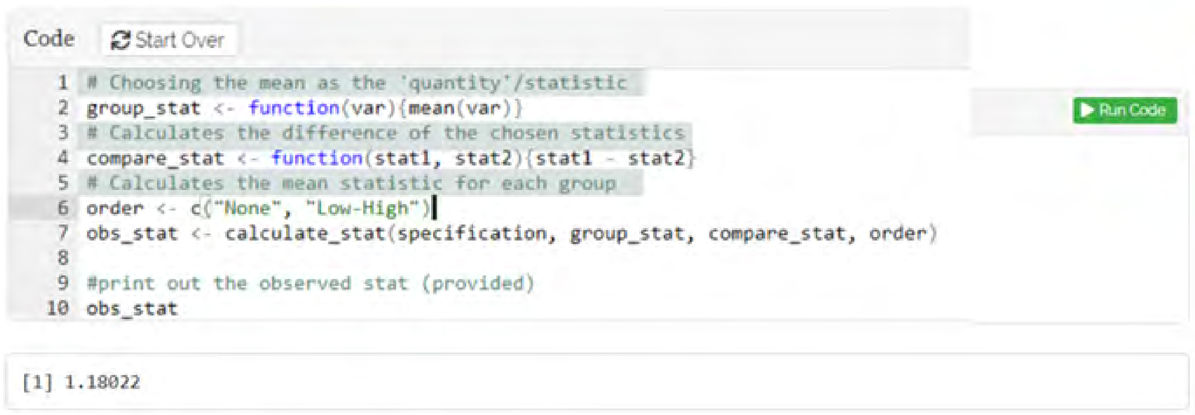
Step 1

```
#
obs_data <- read_csv("https://goo.gl/DhxBYy", col_names = TRUE)

# print out the data as table
obs_data
```

Figure 4.6: Phase 3 Step 1 screenshot and teachers notes alongside Phase 4 instructions and Step 1 code chunk

They added the comment, "Reads in data from csv file and saves it under `obs_data`, with heads as `col_names`" to the code before running the code to produce a table as the output. When presented with the code for Step 5, Ingrid read the code out loud and immediately pointed to Step 5 on the A3 sheet (Figure 4.3) and said, "and this is where it does this!" Ingrid quickly wrote the comment, "Compares the difference from the 1000 simulations to the observed difference and finds the proportion of differences that are at least as big as observed", ran the code and stated, "there's the *P-value*!" (pointing to the output of 0.025).

It was also conjectured that the use of the code-driven tool would develop new *computational knowledge* during this phase. For example, when presented with the lines of code for Step 2, Naomi and Ingrid determined that, whereas *VIT Online* had "guessed" which variable was the explanatory and which was the response, with the code they needed to *specify* these variables themselves. They referred to the A3 sheet annotated with their comments in Phase 3 to make sense of the code. Neither teacher had used the verb *specify* when describing this same step in Phase 3, but it is the name of the function `specify()` from the *infer* package. Similar to the approach in Steps 1 and 2, the teachers read aloud the code for Step 3 to each other first before attempting to make sense of each line. After several rewrites of their comments, they reached agreement on the comments shown in Figure 4.7.



```
 1 # Choosing the mean as the 'quantity'/statistic
 2 group_stat <- function(var){mean(var)}
 3 # Calculates the difference of the chosen statistics
 4 compare_stat <- function(stat1, stat2){stat1 - stat2}
 5 # Calculates the mean statistic for each group
 6 order <- c("None", "Low-High")
 7 obs_stat <- calculate_stat(specification, group_stat, compare_stat, order)
 8 
 9 #print out the observed stat (provided)
10 obs_stat
```

```
[1] 1.18022
```

Figure 4.7: Code provided for Phase 4 Step 3 showing comments written by teachers highlighted and partial output

The process used to agree on these comments revealed some of the reasoning used when the teachers were presented with unfamiliar functions and object names. For example, the teachers quickly matched lines 2 and 4 of the code for step 3, shown in Figure 4.7, to familiar features of the GUI-driven tool, seemingly helped by the *intuitive* object names used of `group_stat` and `compare_stat`. For line 6 of the code, I intended for the naming of the object as `order` to provoke the teachers to realise that unlike *VIT Online*, they needed to define the order they wanted the difference to be calculated. However, Ingrid initially thought that the purpose of this line was to tell the computer which two groups to compare. In response to Ingrid, Naomi explained that the code was telling the computer the order to calculate the difference.

> Naomi: When it [VIT] does it, it always calculates so that you end up with a positive difference
>
> Ingrid: So, the ordering is not by the estimate value, it's by the code

When the teachers began to discuss the function `calculate_stat()`, Naomi said, "I must admit! I'm real confused here." It appeared the use of objects as arguments for the `calculate_stat()` function was the source of the confusion. Ingrid read out the arguments aloud and realised that the `specification` object was defined in an earlier step. The teachers then scrolled up the page to the previous step to see where the object `specification` had been used and what it represented. It appeared that progressive revealing of the code for each step may have hidden the dependent nature of the steps.

It appeared that Phases 2 and 3 of the task supported the teachers' ability to make sense of the unfamiliar code presented to them, as did their pre-existing knowledge of the randomisation test. For example, when presented with the code for Step 4, Naomi immediately made links between each line of code and what she knew about the randomisation test. This may be because the "verbs" `hypothesize` and `generate` are familiar statistical terms, as are the words `null`, `independence`, and `permute`. The teachers appeared to be supported to think computationally during this phase as the only new knowledge to learn was computational.

4.5.5 Phase five: Re-using code chunks to carry out the randomisation test

It was conjectured that this phase of the task would encourage the integration of statistical and computational thinking. Recall that Naomi and Ingrid were given a new data set from a similar experiment, called the *man* data, and asked what they could learn from these data with respect to the motivating question of whether the use of prompts influenced height estimates. The general approach used by Naomi and Ingrid for this phase was to copy the code from each step in Phase 4, change it if necessary, run it to check it worked, and then move on to the next step. Typically when Naomi and Ingrid ran each code chunk, they did not say anything specific about the output produced except to express happiness that it appeared as they expected.

The teachers did demonstrate *some* integration of statistical and computational thinking in this phase, although the statistical thinking utilised predominantly contextual knowledge. There was very little discussion about the features of the data, even though an important part of statistical modelling is to visualise the data being used for a statistical test. For example, when checking that the variable names used in the code for Step 2 matched the variable names in the data set, the teachers opened the data as a spreadsheet to scroll through the data. The teachers focused on looking at the data in terms of *variable information* for their code, not in terms of statistical features. For example, they identified that the two groups were labelled "yes" and "no" in the *man* data but they did not discuss that there were more data, nor that the heights were measured in centimetres in the *man* data not metres as they were in the *giraffe* data.

During this phase, there were examples of development of the teachers' computational thinking. For example, after copying the code for Step 1 from Phase 4, Ingrid suggested changing the name of one of the variables `obs_data` to `man_data` before realising this was not required as, "otherwise we'll have to change stuff later on." Her initial idea to change the wording in the code could reflect the differing requirements of language use within a programming context and a statistical interpretation context. Additionally, in Step 3 as she was copying the code chunk from Phase 4, Ingrid said, “we’re going to have to change something here, because it’s Yes instead of High-Low (sic)”, recognising a need to change the code provided.

Another example of some integration of statistical and computational thinking, with a focus on linking to contextual knowledge, was in Step 3. When the code for Step 3 was executed, the teachers saw that the output was a difference of 4 centimetres, and similar to Phase 2, immediately reasoned in terms of its size and potential causality:

> Ingrid: So, the difference was 4 cm
>
> Naomi: Oh, that’s interesting, not a very big difference
>
> Ingrid: I think what might have helped was the railing in the photo
>
> Naomi: Yeah, I think it was anchored by that [railing] as well as the 180 [referring to the photo branding estimation180]

However, there were no visual graphics produced by the code to support their discussion of the difference of 4 cm, and the randomisation test had not yet been fully carried out. Following this, Steps 4 and 5 were quickly copied and the code executed to produce a tail proportion of 0.239. Naomi stated “so no difference between prompt and no prompt”, which is a common incorrect conclusion from a large *P-value.*

4.5.6 Phase six: Exploring “What if?” changes to the code by creating “new” test statistics

It was conjectured that this phase of the task would further develop the teachers’ integration of statistical and computational thinking. Throughout this phase, as Naomi and Ingrid explored a range of different test statistics, they appeared to develop a greater understanding of the relationship between the test statistic and the re-randomisation distribution of the test statistic. The use of the code-driven tool allowed them to automate the production of new re-randomisation distributions each time they changed the definition of the test statistic.

For example, after using the code provided to carry out a randomisation test with the *giraffe* data using the difference of two means as the test statistic, Naomi then made a suggestion for their exploration:

> One of the things we noticed was that there was a difference in spread, we could look at differences in spread. What if we compare interquartile range? Maybe, because I know you can compare quartiles in VIT, I don't remember if you can compare [IQR].

Ingrid found the code chunk where the test statistic was defined and Naomi suggested changing the names of the function from `median()` to `IQR()`. Neither teacher was sure if there was a function provided by *R* for calculating the interquartile range and the fact that the function is actually named `IQR()` meant they did not need any *code-related knowledge* to modify the code. After making this change to the code, Ingrid ran the code again. The teachers first discussed the output for the test statistic, a difference in IQRs of 1.65 metres, in reference to the dot plot provided on the A3 sheet used during Phase 3, not the plot provided in the output.

The teachers then turned their attention to the re-randomisation distribution produced (see Figure 4.8a).

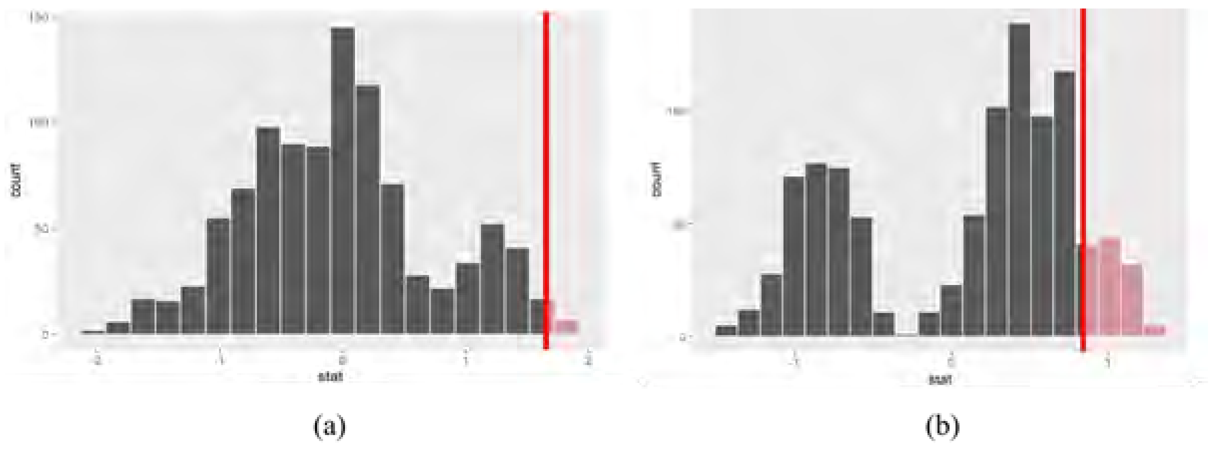


Figure 4.8: The re-randomisation distributions for the difference of interquartile ranges (a) and the difference of standard deviations (b) using the giraffe data

Ingrid commented on the "weirdness" of the re-randomisation distribution and expressed a desire to run the test again, stating, "Yeah, it looks bimodal. I wonder if that's by chance though." She had recognised the dominant feature of the distribution but wondered if it was just a "one-off" artefact of the simulated computational process. Naomi, however, initially reasoned away the "weirdness" of the re-randomisation distribution by first reasoning that, "Yeah, but you get that [distribution] for medians" before partially restating her incorrect understanding about re-randomisation distributions:

> That is interesting because you would expect to see something more … it shouldn't necessarily be normal but it should be roughly symmetric.

Ingrid ran the code and while they waited for the output to be updated, the teachers continued to reason about the shape of the re-randomisation distribution:

> Ingrid: We've got 1000 observations
>
> Naomi: But yeah, sometimes with small samples and medians and things, you don't get normal but…
>
> Ingrid: Because our group sizes are so unbalanced?
>
> Naomi: But it should still be symmetric. There are particular values that can come up, but it should still be symmetric

Ingrid's reflections appeared to be based on explaining why the shape of the re-randomisation distribution of the difference in IQRs is not what was expected by offering thoughts related to specific features of the data used for the randomisation test, whereas Naomi's reflections appeared to be based on general concepts related to an expected sampling distribution. The teachers then checked the re-randomisation distribution for the second attempt, and found that the shape was very similar to the first simulation and still clearly bimodal, with Naomi remarking, "yeah that's odd, that it isn't symmetric because it should be symmetric."

After they noted that the shape for the re-randomisation distribution was consistent across each attempt, Ingrid suggested trying out the standard deviation to see if they still got the bimodal features for the re-randomisation distribution of the difference in standard deviations between the two groups. After a brief discussion with me, initiated by Naomi who wanted to know why the re-randomisation distribution was bimodal, the teachers tested the difference of the two standard deviations and observed the same bimodal shape feature for the re-randomisation distribution produced (see Figure 4.8b). Naomi then considered the test result and the possible influence of the "outlier":

> Interesting that it [the test] shows a difference in the IQRs but not the standard deviation … IQR is going to be more suitable because there's that outlier, so I would tend to use an IQR … but yeah, whatever you do, which one gets this one [the outlier] is going to have a wider standard deviation, so probably better do IQR.

While adjusting code for their next exploration, Naomi and Ingrid linked the idea that, "which one [group] gets this one [the outlier] is going to have a wider standard deviation" to the bimodal feature of the re-randomisation distribution for the difference of the two standard deviations. Naomi and Ingrid then explored the use of a ratio to compare the IQRs and the means of the two groups. Naomi then suggested that they use the *man* data to explore the different test statistics.

Similar to Phase 5, the teachers needed to make changes to the code to adapt to the different data set. The teachers defined the order as `order <- c("no", "yes")`, which resulted in a negative test statistic. When the two teachers discussed the order of the difference calculation for the test statistic, they appeared to utilise an integration of both statistical and computational thinking, which helped to partially resolve their earlier uncertainty about this line of code.

> Ingrid: … because you can decide the order you want to call it in [the code to calculate the test statistic], we could have gone "yes", "no", but if we did "yes" first it [the observed difference between group means] would be positive
>
> Naomi: But if the original data was "no" and "yes" we wouldn't want to swap that .. OK let's swap it back to "yes", "no" and then we'll have a positive thing [referring to the observed difference] … and it's always going to use it [the order] for random reallocation

In the excerpt above, the teachers appear to be drawing on computational knowledge (e.g., having to tell the computer what order to calculate the difference) and statistical knowledge (e.g., which group had the higher mean in the experiment data).

Throughout this phase, the contextual part of statistical thinking was suppressed as the teachers focused on distributional properties and aspects of statistical knowledge related to the randomisation test. Unlike Phases 2 and 5, when the teachers interpreted the results of the randomisation test by linking back to the context, during this phase the focus was on using code to quickly create new visual representations of re-randomisation distributions. At the end of this phase, Naomi and Ingrid were asked to share with the other teachers what they explored.

> Naomi: We did the IQR, the difference and the ratio, and that had a really interesting distribution. Because the sample sizes are unequal it looks quite different than what we are used to seeing. We're used to seeing thing that are more symmetrical because the sample sizes are more equal but with one smaller than the other you get this unsymmetrical distribution.

Note that Naomi now realised that it was possible for the re-randomisation distribution to be unsymmetrical. In fact, her written reflection on concepts she felt she understood better as a result of participating in the research workshops, she said, "re-randomisation distributions are not necessarily symmetric."

4.6 Reflection

One objective of my research was to learn more about how the proposed design principles and considerations could help introduce learners to using code-driven tools for statistical modelling. One of the challenges in designing the learning task was finding ways to reduce the *cognitive load* when integrating different and often new, contextual, statistical and computational ideas. For example, although these teachers were familiar with randomisation tests for the difference of two means or medians, they had not previously explored other test statistics.

It appears that overall the learning task was successful and that this could be due to the specific design principles used when creating the phases: (1) immerse, (2) re-familiarise, (3) describe, (4) match, (5) re-use, (6) explore. The effectiveness of the design principles, which were explicated after the development and implementation of the task, could also be explained because they were based on teaching strategies sourced from both statistics education and computer science education research (e.g., Garfield et al., 2012; Sentance & Csizmadia, 2017). I now discuss these six design principles in light of the results presented, positioning the teachers as the learners.

4.6.1 The six design principles

Aligning to the *immerse* principle, Phase 1 was devoted entirely to familiarising the teachers with the data context and stimulating their interest in the problem. This phase required no statistical or computational knowledge, so allowed all the teachers to participate regardless of confidence with the randomisation test. The data context was then used for all the remaining phases and the data context appeared to support teachers' reasoning (c.f. Weiland, 2016). Aligning to the *re-familiarise* principle, Phase 2 allowed teachers to learn about a new data set, the *giraffe* data, using a familiar tool (*VIT Online*) and a familiar statistical method (the randomisation test). The findings presented for Phase 2 indicated that this approach allowed the teachers to focus on interpreting the randomisation test *contextually* and on the features of the experiment data, in particular the noticeable differences in spread between the two groups.

After building specific knowledge about the experiment data and the randomisation test result using the difference of two means, the *describe* principle demonstrated in Phase 3 prompted teachers to think *computationally* about the randomisation test process. In particular, the teachers were confident describing each step of the randomisation test in computational terms, with Steps 3 and 4 providing some challenge. The *describe* principle appeared to offer similar benefits to the *unplugged* or hands-on activities used within existing learning progressions for simulation-based inference (e.g., Chance et al., 2004). The teachers began Phase 4 knowing what computation was needed for each step and could focus on learning *how code* tells the computer how to carry out each step (c.f. Pruim et al., 2017). The *match* principle demonstrated in Phase 4 appeared to help build the teachers' confidence with using the code-driven tool and introduced some new knowledge about the functions used within the code. As the code itself did not need changing, their unfamiliarity with the functions used did not prevent the teachers from carrying out the randomisation test. Additionally, the teachers frequently accessed and used their pre-existing knowledge of the randomisation test, their specific knowledge of the test result for the difference of two means for the *giraffe* data, and their annotated A3 sheet about the *computational* process during Phase 3.

During Phase 5, the *re-use* principle appeared to lead the teachers to successfully adapt the code to carry out the test with the *man* data, however, decisions made were not informed by a visual representation of the experiment data. Although the focus on "getting the code to work" prompted computational thinking, care will be needed to ensure tasks designed using a code-driven tool do not only reward *text outputs*. In Phase 6, the *explore* principle appeared to encourage the teachers to try out new test statistics for both sets of data and to test their conjecture that the spread of the two groups was different. Their exploration during this phase also led to a partial resolution of an unexpected misconception about the randomisation test revealed early in the task, which was that the re-randomisation distribution will always be symmetric and centred around zero for any test statistic. When the teachers moved to a code-driven tool that allowed them to test other statistics, Naomi and Ingrid began to realise why the re-randomisation distribution may not have a symmetrical distribution by linking the test statistic, the features of the experimental data and the features of the re-randomisation distribution.

4.6.2 Design considerations

The introduction of new knowledge

A design consideration was how to balance the learning of new statistical, computational, data-related and tool-related knowledge within the same learning task. Phases 2 to 4 demonstrate this consideration by only introducing one new idea at each phase. Phase 2 introduced a new data set, Phase 3 the new idea of describing the steps computationally, and Phase 4 new code that matched the steps described in Phase 3. Although the amount of new *R programming* knowledge teachers developed was minimal, since they were provided with code to use, one of the reasons the teachers felt the "code was not so scary" was because of the "chunking" of the randomisation test by computational steps, the unplugged nature of Phase 3, and the revealing of the code for each step progressively in Phase 4. The teachers were not introduced to computational thinking at the same time as learning a new tool. While it is conceivable to design tasks where new statistical, computational, and tool-specific knowledge are built simultaneously, these results appear to highlight a potential risk that the different knowledges may not be gained equally and that gains in one area of knowledge may be at the expense of another area (c.f. Cobb, 1997). For example, through Phases 3 to 6, the teachers' discussion of the data context and the study design were minimal as they were focused on the code-driven tool. The differing nature of learner interactions with GUI-driven and code-driven tools was summarised by teachers in terms of "what the computer needs" versus "what humans need."

Level of computational transparency

It was a challenge in the design process to balance the need for a learner to: (1) access high quality graphical representations of the data and statistical modelling methods used, (2) read and write code that was *computationally transparent*, and (3) develop self-belief and self-confidence in using a code-driven tool. At various times during the code-driven phases of the tasks, the teachers struggled with some of the new *computational knowledge* presented via the code. However, similar to how Erickson et al. (2019) propose the naming of "data moves" to help support learners move between different tools, some of the specific names of functions offered by the *infer* package helped the teachers to make connections with their pre-existing statistical knowledge for the randomisation test. Notably, the teachers often made use of GUI-driven tools during the phases that were designed to use code-driven tools, suggesting that it may be helpful to allow or even encourage learners to "fold back" to familiar GUI-driven tools during the move from GUI-driven tools to code-driven tools.

4.6.3 Conjectured thinking for each phase

In designing a learning task to introduce teachers to code-driven tools, it was conjectured that the thinking would follow a path of statistical (Phases 1 & 2), computational (Phases 3 & 4), to an integration of the statistical and computational (Phases 5 & 6). Although the thinking was observed in all phases, in Phase 5 the teachers' statistical thinking was focused on the contextual aspects and in Phase 6 the teachers' consideration of contextual knowledge was suppressed while they engaged with the underlying structure and properties of the re-randomisation distribution in the statistical domain. This may not be surprising when considering Cobb's (1997) contention that when a person enters deeply into a thinking element, the other elements of thinking become irrelevant or suppressed as the mind has no room for them (cf. Sweller at al., 1998). According to Lee et al. (2011), computational thinking involves abstraction, automation, and analysis. In this study these teachers appeared to show these characteristics while engaging with a *statistical modelling* task. In particular, abstraction was demonstrated by the teachers' generalisations about re-randomisation distributions for different test statistics, automation was demonstrated by the teachers' engagement with the code representation of the randomisation test algorithm and analysis was demonstrated by the teachers' interpretation of the features of the re-randomisation distribution. Hence, I conjecture that the teachers indeed were developing an integration of statistical and computational thinking.

4.6.4 Summary

There is an increased focus at the New Zealand secondary school level to integrate computational thinking across all subjects. As GUI-driven tools dominate the teaching of statistics, new research is needed to better understand how to introduce teachers to using code-driven tools and how an integration of statistical and computational thinking could be supported in a learning task. In line with Biehler (2018), I concur that greater consideration is needed about the relationship that exists between tools, task design and learners' statistical and computational conceptions.

Using the randomisation test as an example of statistical modelling (Task 3), my findings indicate that the six design principles and three considerations could provide an explanation as to why the learning task supported teachers' introduction to code-driven tools and encouraged an integration of statistical and computational thinking. These design principles and considerations represent the first iteration of the Introducing Code-driven Tools (ICT) task design framework. In Chapter 5, I explore the introduction of code-driven tools through a predictive modelling task

(Task 2) and use the task and its implementation with teachers to evaluate and refine the task design framework and examine integrated statistical and computational thinking.

Chapter 5: Predictive modelling task

5.1 Introduction

Teaching recommendations for implementing data science at the high school and introductory tertiary level include: placing greater emphasis on predictive modelling (Biehler & Schulte, 2017; Gould, 2017; Ridgway, 2016); immersing students in data-rich contexts by sourcing dynamic ("live") data from the internet (Engel, 2017; Hardin, 2018); and providing opportunities for students to integrate both computational and statistical thinking (e.g., De Veaux et al., 2017; Gould, 2021). An obstacle to implementing these recommendations at the high school level is teacher content knowledge and computing skills, particularly in the areas of machine learning and the use of computer programming (coding) to access, manipulate and visualise data from sources such as APIs (Application Programming Interfaces).

While materials for teaching data science at the high school level provide examples of curriculum designs and how computational tools can be used (e.g., mobilizingcs.org/introduction-to-data-science, key2stats.com, prodabi.de, idssp.org), there is a need for the explication of the design principles used to develop the learning tasks. Clear guidance is also needed for how to design learning tasks that will successfully engage a wide range of high school students with data science, particularly for those who lack confidence with mathematics and computing (Burr et al., 2021).

Using a DBR approach, I created the first iteration of a design framework to inform the development of new tasks to introduce code-driven tools through statistical modelling (see Chapter 4). In this chapter, I explore the design and implementation of a task for introducing teachers to predictive modelling using dynamic data sourced from an API (Task 2). I consider if the task design framework can be applied to a task that uses a different source of data (dynamic via APIs), a different statistical modelling situation (predictive modelling), and predominantly one type of tool (code-driven). I also explore what statistical and computational thinking practices emerge when the task is implemented with teachers.

5.2 Teaching predictive modelling and APIs

High school statistics courses have traditionally used data collected within formal studies to teach students about study design and statistical inference. Education researchers are re-thinking and expanding their ideas about data and approaches to statistical modelling and have suggested the inclusion of machine learning approaches and associated algorithmic models in high school curricula (e.g., Biehler & Schulte, 2017). From a learning perspective, algorithmic models could offer a more accessible and conceptually simpler mechanism to introduce students to data science than inferential methods (Gould, 2017; Ridgway, 2016), and research by Zieffler et al. (2021) suggested there might be similar benefits for high school statistics teachers. Predictive modelling, with its focus on developing models by learning from features of data to make predictions and forecasts for likely future outcomes, could also provide opportunities for students to integrate both computational and statistical thinking (e.g., De Veaux et al., 2017). Although machine learning approaches to predictive modelling could be used, previous research cautions against using "black box" approaches to teaching modelling (e.g., Biehler & Schulte, 2017; Magana et al., 2011).

Regression models can be used for predictive modelling and simple linear regression is commonly taught at the high school level (e.g., Bargagliotti et al., 2020). However, a different perspective is needed to teach machine learning and algorithmic modelling than the traditional use of linear regression (Biehler & Schulte, 2017). Teachers' existing understanding about linear regression also needs to be considered. The purpose of the linear model can be unclear to teachers, for example, whether the line fitted represents a model for a general relationship or a summary of the data-specific relationship (Casey & Wasserman, 2015). Additionally, little is known about how teachers will reconcile algorithmic modelling approaches alongside traditional modelling approaches within the same teaching programme. Biehler and Schulte (2017) suggested that teaching predictive modelling from a data science perspective could build from familiar understandings of linear regression but include more emphasis on validation through residual analysis and predictive accuracy.

Simple linear regression models are too "simple" to produce high rates of predictive accuracy using point predictions, but more advanced approaches to regression would be beyond the scope of the high school statistics classroom. An informal approach to introducing predictive modelling could draw on the success of informal inference research (e.g., Makar & Rubin, 2018). A key characteristic of using an informal approach is employing visual representations to build concepts and inform decisions, utilising specially designed software such as *TinkerPlots* (Konold & Miller, 2015) and *VIT* (Visual Inference Tools, Wild et al., 2017). An informal approach could be used for introducing predictive modelling, for example, by generating prediction intervals based on a visual estimate for the prediction error and evaluating the model in terms of what percentage of the outcomes appear to be "covered" by the prediction intervals.

Students' experiences with training and testing models, such as cross validation, do not need to be formal. Using data from sources such as APIs provides an opportunity for students to train and test models with different data sets, therefore supporting learning about concepts such as overfitting, underfitting, and generalisability. The use of dynamic data more closely aligns with predictive modelling in modern applications, such as monitoring social media usage or customer interactions on web pages and using past customer transactions to predict future purchasing behaviours. Modern data contexts may also be more engaging for teaching high school students (Gould, 2010; Ridgway, 2016).

There are many computational barriers for high school teachers to use APIs for teaching. Providing "data portals" that allow students to access data from APIs without coding (see Erickson, 2020) are one possibility for opening the world of dynamic data for teaching. However, accessing data from APIs is not the only computational barrier. Using data that is obtained can also raise difficulties. Data formats and hierarchical structures such as JSON (JavaScript Object Notation) and XML (Extensible Markup Language) are not common for teachers, nor are manipulations with the raw data obtained from the API, for example, working with timestamps or text. Recommendations to use interactive documents such as *RMarkdown* (Allaire et al., 2021) to access dynamic data (e.g., Hardin, 2018) are not likely to be widely adopted at the high school level as they involve installing and using specialist software. Teachers may prefer tools that are web-based, free, require minimal time to learn how to use, and offer collaboration and easy sharing of tasks (Biehler, 2018). Therefore, care is needed to design predictive modelling tasks that balance new statistical and computational ideas, with the learning demands of using new data technologies.

A common thread to discussions about data science education is that students need to integrate both statistical and computational thinking (e.g., De Veaux et al., 2017; NASEM, 2018), and that students need to develop at least some computer programming skills (e.g., Cetinkaya-Rundel & Rundel, 2018; Nolan & Temple Lang, 2010). Arguments exist for and against teaching statistics at the high school level using *code-driven* tools, computational tools which users interact with predominantly by entering and executing text commands (code). Statistics education research at the high school level, however, has largely involved *GUI*-driven tools, computational tools which users interact with predominantly by pointing, clicking, or gesturing.

With respect to introducing predictive modelling and data sourced from APIs, using a code-driven tool could support the teaching of statistical and computational thinking. Central to statistical thinking is the use of statistical models (Wild & Pfannkuch, 1999). Using a code-driven tool could lower the cognitive demands of the statistical modelling task (Son et al., 2021), as code can be used to articulate modelling steps (Kaplan, 2007). I contend that computational transparency, that is, how obvious the computations performed by the code are to the learner, is an important consideration when using code-driven tools for teaching statistical modelling. By using a code-driven tool, students can modify requests when accessing and using data from APIs, allowing the data context to be central to their learning and decisions (Weiland, 2016; Wild & Pfannkuch, 1999). Furthermore, the use of a coding approach has the potential to allow for exploration of "what if?" scenarios, explorations that are often restricted by the options provided by GUI-driven tools.

The design of the task should take cognisance of the presentation and interface for the computational tool and allow students to tinker with a model articulated with code and to visualise changes instantaneously (cf. *Tinkerplots*). The *R* (R Core Team, 2020) package *learnr* (Schloerke et al., 2018), for example, provides a way to produce an interactive web-based task where students can execute small "chunks" of *R* code within a web browser. Since the release of *learnr* in 2017, the package has had over 250,000 downloads but there are very few research articles that describe the use of the tool to design learning tasks (e.g., Wiedemann et al., 2020). In general, it is difficult to find substantial literature that explicitly communicates strategies for designing tasks that introduce code-driven tools for statistical modelling at the high school level.

5.2.1 The New Zealand teaching context

New Zealand has one national curriculum which is taught at nearly all high schools. Grade 12 statistics students are assessed against the curriculum using school-based assessment tasks and national examinations. Curriculum and assessment materials provided by government educational agencies (e.g., NZQA, 2019) were used to review current approaches to the teaching and assessment of predictive modelling. I identified that Grade 12 statistics students are expected to use linear regression models to make predictions but are not required to engage with sample-to-population inference ideas when using linear regression models, for example, interpreting confidence intervals for the model parameters. Students are also not expected to: use a prediction model developed with one set of data to generate predictions for cases within a different set of data; generate prediction intervals from a model; evaluate a model in terms of predictive accuracy; discuss precision versus accuracy; access APIs as a source of data; or use computer programming as part of the predictive modelling process. Consequently, high school statistics teachers are unlikely to have experience with designing and implementing tasks that employ these approaches.

5.3 Task design framework

Through the retrospective analysis of the randomisation test task, I explicated a design framework to construct tasks that introduce learners to code-driven tools through statistical modelling (Chapter 4). Because of the iterative nature of design-based research, I evaluated the first iteration of the design framework (Section 4.3) against the criterion of producing a more general design framework rather than one that was too specific to the nature of the tools and tasks used previously.

The learning task from the first iteration was constructed to move learners from a familiar GUI-driven tool to an unfamiliar code-driven tool for carrying out the randomisation test, a familiar statistical modelling approach. Consequently, some of the design principles specifically mentioned GUI-driven tools and assumed existing knowledge of the statistical modelling approach.

For the *second iteration* of the task design framework, the following changes were made: the references to GUI-driven tools were replaced with references to statistical modelling ideas or processes; a new design consideration concerning the tools used was added; and the descriptions of the design principles and considerations were modified. An overview of the *second iteration* of the design framework discussed in this chapter is shown in Figure 5.1.

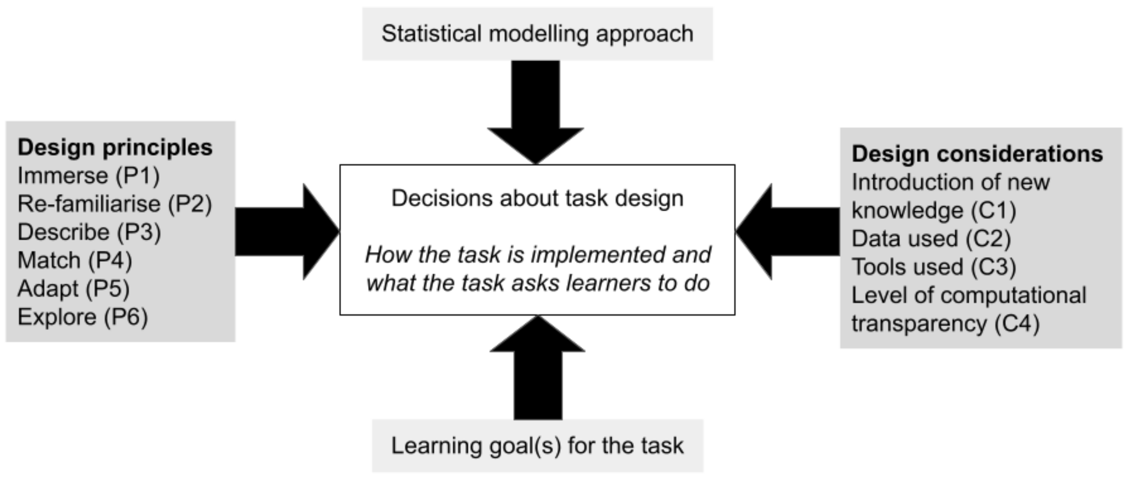

Statistical modelling approach
Design principles
Immerse (P1)
Re-familiarise (P2)
Describe (P3)
Match (P4)
Adapt (P5)
Explore (P6)
Decisions about task design
How the task is implemented and what the task asks learners to do
Design considerations
Introduction of new knowledge (C1)
Data used (C2)
Tools used (C3)
Level of computational transparency (C4)
Learning goal(s) for the task

Figure 5.1: Design framework for constructing statistical modelling tasks that introduce code-driven tools

To use the design framework to construct a task that introduces a code-driven tool for statistical modelling, the task designer needs to decide what statistical modelling approach will be used. There will be an initial idea about the learning goal(s), which are shaped during the design of the task and finalised at the end of the construction process. The design principles are used to inform decisions about features of the learning task in terms of specific actions or experiences for learners and the chronological order of these actions or experiences. The design principles *immerse*, *re-familiarise*, *describe*, *match*, *adapt* and *explore* are presented in Table 5.1.

Table 5.1: The six design principles of the design framework

Design principle	**Learning action or experience**	**Anticipated learning**
Immerse in data context (P1)	Participate in activities that promote engagement with the data context	Understanding the nature of the data that is used across the task
Re-familiarise with statistical modelling ideas (P2)	Carry out familiar statistical modelling activities without using code	Application of statistical thinking
Describe computational steps of statistical modelling process (P3)	Use words to describe key computational steps of statistical modelling process	Decomposition of modelling steps and recognising required computation
Match statistical modelling steps to code chunks (P4)	Read and match lines of code with statistical modelling steps	Recognising aspects of code syntax and structure
Adapt code chunks with slight modifications (P5)	Identify features of code to change to complete a statistical modelling action	Integration of statistical and computational knowledge
Explore "what if?" changes to code (P6)	Modify at least one aspect of provided code to produce new or unexpected outputs	New knowledge gained by integrating statistical and computational thinking

Alongside the design principles, the construction of the task is simultaneously guided by four design considerations that inform broader decisions across the learning task: the *introduction of new knowledge*, the *data used*, the *tools used* and the *level of computational transparency*.

- The *introduction of new knowledge* (C1) refers to identifying content and using a sequence of phases and steps within a task to introduce learners to new ideas.

- The *data used* (C2) refers to selecting and using different variables or different sets of data within the same data context for the task.

- The *tools used* (C3) refers to combining different tools for statistical modelling (unplugged, code-driven, GUI-driven) and connecting actions or representations between tools within the task.

- The *level of computational transparency* (C4) refers to how obvious the computations performed by the tool are to the learner.

5.4 Task design characteristics

I now discuss how the *second iteration* of my task design framework aligns to the task used in this chapter (see Appendix D). The statistical modelling approach was an informal method that relied on teachers' reasoning with features of visualisations to construct a model that generated prediction intervals. The form of the prediction model was `predicted y = a + bx ± error` , where `a` and `b` are the y-intercept and slope of a linear model respectively, and `error` is a numeric value visually estimated by the teachers to model prediction error. The learning goal for the task was for the teachers to create a model that generated prediction intervals, using data sourced from an API.

5.4.1 Data-related design decisions

The decision to use dynamic data from an API for the task was informed by the design consideration of *the data used* (C2), as well as the larger research study goal to provide a data science perspective for statistical modelling. The use of dynamic data provided an opportunity to learn about a new modern data source and related computational ideas, at the same time as supporting development of predictive modelling ideas by providing different data sets. Conventionally, the process of training and testing a prediction model would involve the same set of data, for example, randomly allocating 80% for training and 20% for testing. However, as the data for this investigation was sourced from an API, it was decided to ask teachers to develop a model based on data from one search query and apply it to data from a different search query. The approach also allowed for similar reasoning one might face when using an inference made from a sample from one population and applying this generalisation to another population or inferring to a wider population.

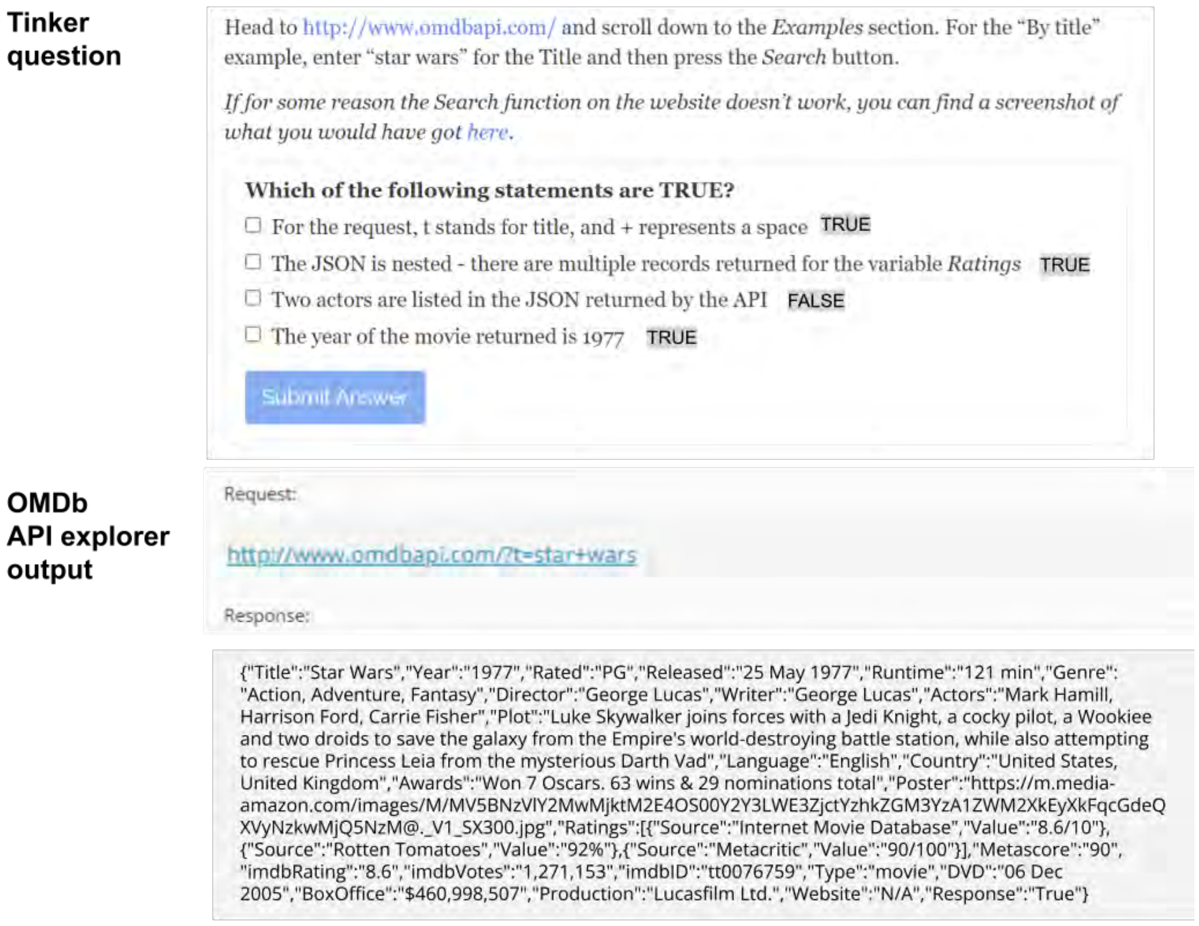
The OMDb API (omdbapi.com) was chosen because it provided data about movies, including information about movie ratings from three different sources. Additionally, OMDb provided an "API explorer", a graphical-user interface (GUI) to send requests to the API, without using a programming language, a feature that aligned with the design consideration of the *tools used* (C3). Specifically, it was decided to design questions that supported teachers to connect actions and representations between the GUI-driven "API explorer" and the code-driven tool. To support an informal approach to predictive modelling, facilitate access to dynamic data from an API and a code-driven tool, it was decided to design a web-based task.

5.4.2 Web-based task design decisions

The task was implemented using an interactive web page created using *RMarkdown* (Allaire et al., 2021), the *R* package *learnr* (Schloerke et al., 2018) and the tidyverse (Wickham, 2017) ecosystem of *R* packages. The *learnr* package provided many features that aligned with my design framework, including being able to: create individual steps for the task containing instructions, multimedia, and links to other webpages; run small "chunks" of *R* code within the task; embed interactive quiz questions within the task; and reveal each step of the task individually as the task progressed.

When developing the code provided to teachers, the statistical modelling approach for the task and the *level of computational transparency* (C4) were considered. The computations related to accessing data from the API were hidden from the teachers by using a function I developed. To support teachers' understanding of the prediction model, the computations related to linear regression and representing the prediction model graphically were revealed to the teachers. Code was developed using the *R* package *ggplot2* (Wickham, 2016) and provided to the teachers using code chunks. The code chunks facilitated fitting simple linear regression models, quantifying the average size of the prediction errors, tinkering with the model parameters, visualising the prediction intervals, and training and testing a prediction model on different sets of data.

The quiz question functionality of *learnr* was used to construct what I call "tinker questions." A *tinker question* presents a set of related TRUE/FALSE statements that are deliberately written to require action by the learners *within a computational learning environment* in order to evaluate each statement. As opposed to quiz questions that assess existing or recently acquired knowledge, these *tinker questions* were used to stimulate the development of new computational knowledge by encouraging learners to make connections between actions and representations of the *tools used* (C3). Figure 5.2 presents a *tinker question*, which was used for step 1 of the task, along with a screenshot of the expected output from using the OMDb API explorer.


Tinker question

Head to http://www.omdbapi.com/ and scroll down to the *Examples* section. For the "By title" example, enter "star wars" for the Title and then press the *Search* button.

If for some reason the Search function on the website doesn't work, you can find a screenshot of what you would have got here.

Which of the following statements are TRUE?

- ☐ For the request, t stands for title, and + represents a space TRUE
- ☐ The JSON is nested - there are multiple records returned for the variable *Ratings* TRUE
- ☐ Two actors are listed in the JSON returned by the API FALSE
- ☐ The year of the movie returned is 1977 TRUE

OMDb API explorer output

Request:

http://www.omdbapi.com/?t=star+wars

Response:

```
{"Title":"Star Wars","Year":"1977","Rated":"PG","Released":"25 May 1977","Runtime":"121 min","Genre":
"Action, Adventure, Fantasy","Director":"George Lucas","Writer":"George Lucas","Actors":"Mark Hamill,
Harrison Ford, Carrie Fisher","Plot":"Luke Skywalker joins forces with a Jedi Knight, a cocky pilot, a Wookiee
and two droids to save the galaxy from the Empire's world-destroying battle station, while also attempting
to rescue Princess Leia from the mysterious Darth Vad","Language":"English","Country":"United States,
United Kingdom","Awards":"Won 7 Oscars. 63 wins & 29 nominations total","Poster":"https://m.media-
amazon.com/images/M/MV5BNzVlY2MwMjktM2E4OS00Y2Y3LWE3ZjctYzhkZGM3YzA1ZWM2XkEyXkFqcGdeQ
XVyNzkwMjQ5NzM@._V1_SX300.jpg","Ratings":[{"Source":"Internet Movie Database","Value":"8.6/10"},
{"Source":"Rotten Tomatoes","Value":"92%"},{"Source":"Metacritic","Value":"90/100"}],"Metascore":"90",
"imdbRating":"8.6","imdbVotes":"1,271,153","imdbID":"tt0076759","Type":"movie","DVD":"06 Dec
2005","BoxOffice":"$460,998,507","Production":"Lucasfilm Ltd.","Website":"N/A","Response":"True"}
```

Figure 5.2: Example of tinker question used for step 1

The stem of the *tinker question* required the teachers to use the OMDb API explorer to send a request to the OMDb API. By searching for a movie with the title "star wars", both the URL request needed to make the request programmatically and the response to the request as JSON were generated. Each of the TRUE/FALSE statements presented for the *tinker question* was designed to help teachers to notice specific features of the output generated. Specific types of statements included those that were FALSE to create conflicting representations and those that encouraged a focus on recognising patterns or structure within code or other computational representations. The teachers then submitted their answer, based on ticking which statements they believed were TRUE. If any of their answers were incorrect, the teachers then needed to repeat the process to identify which statements were TRUE or FALSE. The process of evaluating each TRUE/FALSE statement requires noticing and reflecting on the product(s) of each action, which I conjecture facilitates micro interrogative cycles (Wild & Pfannkuch, 1999).

The design consideration of the *introduction of new knowledge* (C1) led to the decision to use the *progressive reveal* setting for the *learnr*-constructed task. The steps of the task were revealed to the teachers progressively; when they completed each step, the next step appeared below the previous step(s) on the same web page. The *progressive* reveal setting also prevented the teachers from moving to the next step until the *tinker question* was successfully answered. Similarly, this setting prevented the teachers from moving to the next step until the code provided was executed at least once. These settings were used to carefully sequence the order and amount of new computational and statistical ideas introduced across the steps of the task.

5.4.3 Summary of task phases and steps

A summary of the two phases of the task and how they relate to the six design principles and the steps of the task is provided in Table 5.2.

Table 5.2: A summary of the phases, design principle and steps used for the task

Phase	Summary of phase	Principle	Steps
1	Introduce dynamic data from an API	Immerse (P1)	1 to 5
2	Introduce predictive modelling ideas	Re-familiarise (P2), Describe (P3), Match (P4), Adapt (P5), Explore (P6)	6 to 14

Appendix D provides a static version of the full web-based task used, and links to the *R* code used to create the task and a demo version of the task. Each phase and step of the task is now described and includes further explanations about how the design framework is aligned to design decisions.

The first phase focused on *immersing* (P1) teachers in dynamic data from an API, specifically movie ratings from the OMDb API. Tinker questions were used for four of the five steps of this phase (Appendix D: Q1, Q3, Q4, Q5). After being introduced to the structure of API requests and JSON in Step 1, Step 2 built further knowledge of the API by asking teachers to familiarise themselves with selected aspects of the API documentation. The knowledge introduced in Steps 1 and 2 was then used in combination with the code provided in Step 3 to modify requests to the API. Steps 4 and 5 encouraged teachers to adapt the code provided to change the API requests. These steps also allowed the teachers to further familiarise themselves with the nature of the data available about movies from the OMDb API, including the structure of the data and how the data could be manipulated to create new variables.

The second phase focused on introducing teachers to predictive modelling ideas, by drawing on familiar ideas of simple linear regression, notably the intercept, slope, and ideas of sampling variability. Discussion prompts were used across the questions in this phase to facilitate discussion between teachers and to stimulate thinking. Step 6 used a prompt asking teachers to discuss whether they expected there would be relationship between the Metascore and IMDb ratings for movies. Step 7 provided the complete code to create a scatter plot, but the `x` and `y` variables were incorrect, requiring a small change to produce the expected visual representation.

Steps 8 and 9 introduced teachers to fitting simple linear regression models using *R* code and interpreting relevant features of scatter plots. As it could not be assumed that teachers had constructed prediction intervals before, either informally or formally, these steps were used to *re-familiarise* (P2) teachers with the key computational steps required for an informal approach. In particular, Step 9 was used to shift the focus to prediction and to informal approaches for creating prediction intervals using visual features of the data and fitted line. The step specifically asked teachers to discuss whether "You can use an interval to give the predicted metascore rating based on the variation (vertical scatter) observed in the data/plot" and what two numbers they could use for a prediction interval of the `metascoreRating` of a movie that had an `imdbRating` of 7.

Step 10 presented the model to generate prediction intervals, and teachers were asked to estimate the error term for the model. The teachers needed to read the comments within the code that *described* (P3) the lines of the code and then *adapt* (P5) the code, by adjusting the values assigned to the y-intercept, slope, and error. The guidance provided to teachers about how to decide on the numeric value for the error was to base it on their visual estimate of how far away the points sat vertically from the line. Teachers were expected to use the visual representation of the prediction intervals generated — the line fitted and the yellow shaded band around this line — and to *match* (P4) these visual representations to the code used for their model. The discussion prompts in Step 10 were used to encourage teachers to think about model accuracy, by contrasting the percentage of correct predictions using point estimates (the fitted line only) with using prediction intervals (the model developed).

Step 11 asked teachers to discuss how well they thought their prediction model would work with movies that have the word "war" in their title. This step signaled a shift in the task to consider using the prediction model with a new set of data and introduced ideas of training and testing. Step 12 asked teachers to *adapt* (P5) code to *match* (P4) the model they developed in Step 10 and then to discuss how well their prediction model worked for movies with the word "war" in their title. To further explore the idea of generalisability and to provide another visual example of applying their prediction model to a new set of data, Step 13 provided teachers with code to generate and use their model with movies with the word “love” in their title.

Similar to Step 7, the code provided produced a visualisation but there were a few more aspects of the code that needed *adapting* (P5) before the model could be evaluated using the new set of data. Step 14 provided teachers an opportunity to go back to any previous step and look more closely at the code used. This step also provided encouragement for teachers to *explore* (P6) changes to the code by following their own curiosity.

5.5 Analysis

The implementation of the predictive modelling task took place during the second day of the professional development workshops. This was the first task where the teachers were working through a statistical modelling task entirely using a code-driven tool and took place in the afternoon. The theme for the workshop was Star Wars and in the morning session teachers explored a Star Wars API (swapi.dev) by modifying URLs and finding information from the JSON returned (see Fergusson & Wild, 2021). The teacher pairings for this task were: Amelia and Ingrid, Alice and Naomi, and Harry and Nathan (pseudonyms have been used).

All the teachers discussed the specific data context of movie ratings at various times throughout the task and developed new computational ideas related to APIs, including identifying features of JSON, modifying API queries, and using *R* code to access and visualise data from an API. Drawing on familiar ideas related to simple linear regression, the teachers were able to develop a model to generate prediction intervals by adapting code. They also demonstrated some understanding of training and testing a prediction model on different sets of data. I now present in more detail the results of the teachers’ interactions with the task and the consequent emergent thinking that was observed.

5.5.1 Phase one: Introduction to dynamic data from an API

The focus for the task initially was immersing teachers in the data context of movie ratings, by integrating new computational knowledge related to APIs with familiar statistical knowledge including rectangular data sets and features of scatterplots (see Appendix D, Steps 1 to 5). I observed that the teachers were able to read information about movies presented as JSON, even when it appeared they were unfamiliar with the associated vocabulary. For instance, when reading the statements for the Step 1 tinker question, Harry repeated the word "JSON" several times and Alice asked, "What's a JSON, is that the green thing?", indicating both teachers were unfamiliar with the word. The teachers then looked for "JSON" within the OMDb API documentation and successfully negotiated a new understanding that a "JSON" was a type of data structure or new computational representation. The teachers were able to identify the information required within the JSON to answer questions. To illustrate, when considering the statement that referred to the JSON being nested, Amelia responded, "Yes, it's got the little square brackets", demonstrating a connection between the word "nested" to this structural aspect of the computational representation.

The teachers demonstrated new ideas related to the structure of API queries, including the use of parameters and URL encoding. The sequencing of the questions, in particular introducing and progressively revealing new ideas at each step and re-using these new ideas in later questions, appeared to help to build these computational ideas. For example, when Harry and Nathan reached Step 3, the first statement they evaluated as TRUE or FALSE was, "There are around 728 (TV) series with wars in their title." Both teachers remembered from the previous step that there was a way to change the request and pointed to the type=movie part of the query request, before reviewing the OMDb API documentation to confirm the request needed to be changed to type=series. Although Step 3 reminded the teachers to use a "+" to represent a space for a search request, all the teachers referred back to knowledge gained from Step 1, when they used the API explorer to generate JSON for a search for "star wars". Thus, they connected actions and representations between the GUI-driven API explorer and the code-driven tool.

In Steps 4 and 5, where the teachers were provided with code that produced data about the movies in the form of an interactive table, they successfully modified the code and identified information about the number and nature of the variables available. For example, the teachers discovered the function `getResults()` will only return a maximum of 100 results and that data represented using JSON can also be represented as a data table. The task did not ask teachers to manipulate any of the variables provided in the datasets returned from the OMDb API. However, when answering the tinker question used for Step 5, the teachers developed new knowledge about the required computational actions for using other variables from the API data source. For instance, when considering whether the variable Runtime was numeric, Amelia said, "at the moment it's not, you would have to remove 'min'." Similarly, when discussing the variable Genre, Alice commented that each movie has "got multiple genres" and that the variable could not be used to focus on movies from just one genre.

5.5.2 Phase two: Introduction to predictive modelling ideas

The focus of the task then moved to predictive modelling, in particular, the development of an informal model to generate prediction intervals by extending familiar ideas of simple linear regression models (see Appendix D, Steps 6 to 14). The teachers demonstrated they could quantify the size of the prediction errors using an informal visual approach that applied statistical thinking. To illustrate, in Step 9 the teachers were instructed to run the code provided, which produced a scatter plot with the least squares regression line fitted and were asked if the following statement was true: *You can use an interval to give the predicted metascore rating based on the variation (vertical scatter) observed in the data/plot.* The statement was specifically included to support teachers to think about predictive precision. Step 9 also included a discussion prompt asking teachers to create their own prediction interval for the `metascoreRating` of another movie that had an `imdbRating` of 7. To determine their prediction interval, Nathan and Harry placed their pens on top of the computer screen and moved these up and down in parallel positions to the fitted line. Figure 5.3 shows Nathan's red pen in its final position and Harry's blue pen, which was still being positioned.

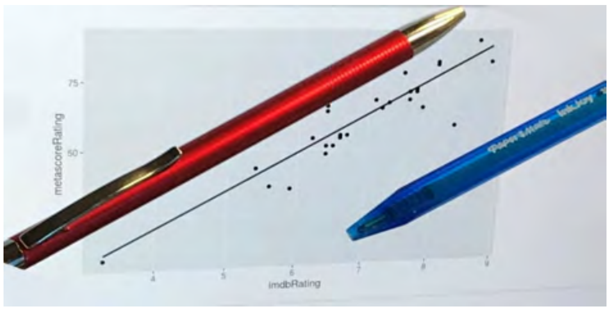
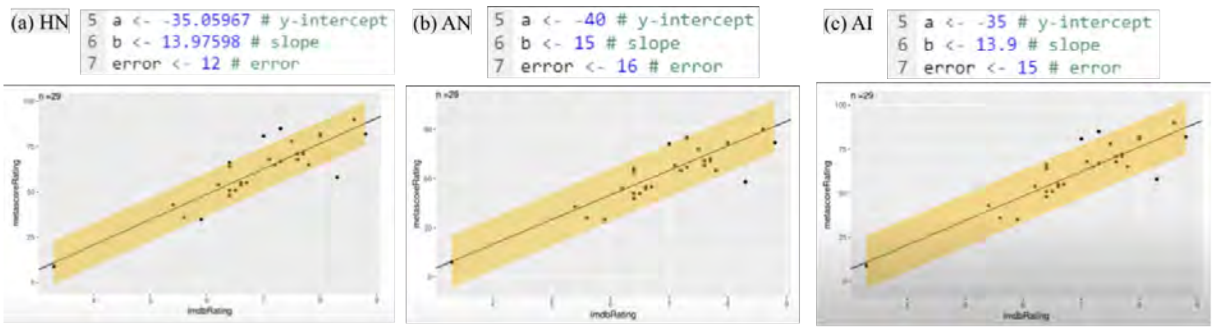


Figure 5.3: Nathan's red pen and Harry's blue pen being used to help determine their prediction interval

To help Harry position his pen, Nathan told him to, "Take out that bottom one! We're not using all the dots, are we?" Harry and Nathan then discussed how much "margin of error" they needed by estimating the vertical distance "above" and "below" the fitted line to their respective pens, settling on half a "square" or "12.5, around 12". Alice and Naomi used similar reasoning, but without pens, which was influenced by the scale breaks used for the `metascoreRating`, leading to the same prediction error value of 12.5.

Step 10 of the task required teachers to develop a prediction model using search data for movies with the word "star" in the title. Although the form of the prediction model was unfamiliar to the teachers, they were all able to adapt the code provided by changing the values for `a`, `b`, and `error`, thus demonstrating some understanding of how the code syntax and structure related to the prediction model. Each pair of teachers took a different approach to developing their prediction model, with their final attempts shown in Figure 5.4.



(a) HN
```
a <- -35.05967 # y-intercept
b <- 13.97598 # slope
error <- 12 # error
```

(b) AN
```
a <- -40 # y-intercept
b <- 15 # slope
error <- 16 # error
```

(c) AI
```
a <- -35 # y-intercept
b <- 13.9 # slope
error <- 15 # error
```

Figure 5.4: The prediction models developed by each pair of teachers for step 10 of the task

Harry and Nathan (HN, Figure 5.4) copied the coefficients for the fitted line, noticing that these were generated from the code provided for Step 10. When reading the line of code `error <- 3`, Harry said, "Error is 3, don't know what that means", and Nathan reminded him of Step 9 and the "margin of error", leading them to use the Step 9 error value of 12 for their model. Alice and Naomi (AN, Figure 5.4) decided on `a` and `b` values for their model entirely "by eye" rather than using the coefficients for the fitted line, which they didn't notice in the output. Alice was unsure about the value for the error, asking Naomi, "Can we play around with that as well?"

The teachers appeared to be guided by model accuracy when developing their informal prediction model. For example, Alice and Naomi initially started with the error used for Step 9 but increased the error value incrementally, with Naomi commenting "If we want 95%, we probably want it a little wider." While Alice and Naomi specifically discussed how many movies/points their prediction model "caught", the other two pairs of teachers only considered the accuracy of their prediction model after reading the discussion prompt for Step 10. For example, Amelia stated in response to the prompt, "If you just use the line, you're basically all wrong, if you use the model we're about 90%."

The learning benefit of attempting to quantify the prediction error for the model first, before using code to visualise the error, can be illustrated in Amelia's and Ingrid's initial response to Step 10. Because they did not read the discussion prompt in Step 9, they did not create a prediction interval before reaching Step 10. Consequently, Amelia and Ingrid needed help from me to begin the development of their prediction model in Step 10, as they were not able at this point in the task to integrate statistical and computational knowledge. There appeared to be too much new information for them to process in one step, as they had to grapple with new code, a new visualisation, and a new type of model at the same time.

The focus on prediction intervals also appeared to support the teachers to consider the purpose of a prediction model. For instance, when the teachers were asked at the end of the task, “In general, what do you want out of a prediction interval?”, Naomi and Harry had the following exchange:

> Naomi: Well, you want it [the prediction interval] to be narrow but you also want it to be realistically narrow. It’s no good saying you want it to be narrow if it doesn’t actually predict very well.
>
> Harry: The thing is, I came back to the question, who is actually going to be using this. If you get a prediction interval of 24 overall it is almost a little bit meaningless, it’s one quarter of 100.
>
> Naomi: Depends if you want to have a precise prediction, then yeah, you need a narrow interval for predicting a number but if you want to capture the variation then a wide interval is useful for communicating the variation.

In this exchange, Harry wanted a prediction interval that is *precise* or narrow and that can be used by someone to make a meaningful prediction, whereas Naomi considered the difference in modelling motivation between *explanation* and *prediction*.

The teachers were asked in Step 11 to discuss how well their model would work for movies with the word “war” in their title. They were given the opportunity to make changes to the code for their model, but none changed the y-intercept or slope for their model. Harry and Nathan kept the error value at 12, whereas Alice and Naomi decided to increase their error to 20, reasoning there would be more variation for the `imdbRating` scores for war movies. Amelia and Ingrid kept their error at 15 but only after a very long discussion on how males dominated the IMDb rating data. Later in the workshop, Amelia and Ingrid stated that they were unsure about whether they should have changed their model based on their contextual discussions or just used the features of the training data. Amelia reflected, “When building your model, and you know your training data is going to be quite different, do you leave it, or do you just make the error really big and be like ‘it works’?” Hence, for four of the teachers, contextual considerations were an important part of the modelling process.

The teachers demonstrated awareness of an iterative approach to modelling, including the benefits of training and testing when developing a prediction model. Having easy access to new data sets via the API supported teachers to refine their informal prediction model and to appreciate the need to consider how well the model might work for new unseen data, as well as the need to consider uncertainty in the data and model. For example, Amelia made the following reflection:

> I think it [the approach] makes the model more meaningful and the understanding about the uncertainty in it [the model], that makes that really realistic. Because the idea is, if you have all the data that you need right there, putting the model to it, you know how well it works for the data that's there. But if you are actually going to use that model to then make a prediction for something that you don't know as much about then 'oh my gosh!' you have to worry about uncertainty.

When given the opportunity to test their model using two different sets of data in Steps 12 and 13 (movies with the word "war" in their title, then "love"), all pairs of teachers successfully adapted the code provided so that the data was appropriate. Additionally, all pairs of teachers modified the error value in response to the accuracy of the predictions. The modifications to the error value were based on considering how many movies/points were captured by the yellow band that represented the prediction intervals produced by the model. It was not intended that the teachers changed their models when using a different set of data, as the task aimed to introduce teachers to the predictive modelling approach of training a model on one set of data and testing the *same model* on a new set of data. However, it appeared that by accessing different data sets and making changes to their prediction models, the task seemed to support ideas about generalisability, as can be seen in Naomi's reflection:

> Actually one of the things that impressed me was that we didn't change the gradient or the intercept and yet that line fitted everything we tried pretty well. It pretty much went through the points no matter which thing we tried it on which was really good.

Step 14 of the task was designed to encourage teachers to explore changes to the code with respect to the data and models, but all teachers did this as part of Step 13, when they were asked to test their model for movies with the word "love" in their title. For example, Naomi tried out searches for movie titles using other words without prompting from the researcher.

Overall, the teachers appeared to develop new statistical and computational ideas related to APIs and predictive modelling, which seemed to be influenced by the provided code-driven learning environment. The design of the web-based task meant that they could simultaneously: become familiar with the specific data-context of movie ratings; use *R* code to access and visualise data, as well as generate and visualise prediction intervals; and train and test a prediction model on different sets of data.

5.6 Reflection

One objective of my research was to identify the statistical and computational thinking practices related to APIs and predictive modelling that could be observed as teachers interacted with a task that was aligned to my task design framework. I observed that all the teachers were able to use a code-driven tool to interact with APIs and to develop a model that generated prediction intervals. I contend the task provided a stimulating "first exposure" to predictive modelling for the teachers, which could be due to the specific design decisions made when constructing the task.

I now discuss specific design decisions, namely, the informal approach to development of a prediction model and the use of: dynamic data from an API; progressive reveal; code chunks; and tinker questions. I make tentative links between these decisions and the results presented about teachers' emergent thinking practices and reflect on my task design framework.

Design decisions

It appears that the informal approach to developing a prediction model, primarily the visualisation of error, provided an opportunity for teachers to develop their own reasoning for what made a good prediction model. Consistent with statistics education research concerning informal inference, the teachers developed their own rules for "making calls" when they decided the value of the error for their prediction model (e.g., Makar & Rubin, 2018; Fergusson & Pfannkuch, 2020). Additionally, the visualisation of prediction intervals appeared to support teachers to assess their models in terms of both predictive precision and accuracy. Although specific concepts of underfitting, overfitting and generalisability were not explicitly discussed by the teachers during the task, Naomi's reflection about keeping the slope and intercept of the prediction model fixed indicated that the task provided a foundation for further development of these predictive modelling ideas.

The use of movie ratings data obtained directly from an API appeared to support the teachers' development of predictive modelling ideas across the task (cf. Weiland, 2016). Using their prediction model with a new set of data to generate prediction intervals appeared to help teachers become familiar with the ideas of data for training and data for testing, a necessary focus according to Biehler and Schulte (2017). The use of an API to access and use more than one data set as part of the modelling process, rather than just one static data set, also appeared to help teachers appreciate the predictive goal of the modelling task which, as Amelia said, was "to use that model to then make a prediction for something that you don't know" (cf. Casey & Wasserman, 2015).

The number and scope of new ideas related to predictive modelling and APIs introduced to teachers within the task was not trivial. The decision to use a web-based task, created using the *R* package *learnr,* provided the important feature of progressively revealing each step visually. To minimise cognitive load, the web-based tasks comprised a sequence of steps that carefully ordered the introduction of new statistical and computational ideas (cf. Wouters et al., 2008). The use of small chunks of code that only required small modifications allowed teachers to engage with new computational ideas such as creating scatterplots, similar to the findings of Wiedemann et al. (2020) from research involving high school students using *learnr* to explore mathematical modelling. In alignment with Son et al. (2021), I argue that the use of *R* code in the task was germane load (Sweller et al., 1998), potentially because the computations represented by the code were familiar to the teachers. By considering the relationship between the tool, task and thinking (e.g., Biehler, 2018), I used code in the task to enable teachers to explore changes to their model and to visualise these changes instantaneously, thus potentially enhancing their statistical thinking (e.g., Ben-Zvi, 2000).

The design and use of tinker questions appeared to support the introduction of new computational ideas. As teachers only needed to focus on one TRUE/FALSE statement at a time, I observed that learning about new computational representations or actions was reduced into "bite sized" interactions. By connecting actions or representations between GUI-driven and code-driven tools, the teachers appeared to develop new understandings of using APIs, for example, features of data structures such as JSON. The interactive approach employed by the tinker questions to learning new computational knowledge could also support students to become active participants in learning from modern data (e.g., Gould, 2010). Furthermore, the tinker questions could help develop *data habits of mind* (see Finzer, 2013) as they provided guidance for noticing and considering selected features of computational representations or actions.

Task design framework

The specific design decisions to use dynamic data from an API, progressive reveal code chunks and tinker questions for the web-based task can be explained by the design considerations of my framework: the *introduction of new knowledge* (C1), the *data used* (C2), the *tools used* (C3), and the *level of computational transparency* (C4). The design principles of my framework (see Table 5.1) also explicate the construction of task phases and steps. Using the *immerse* (P1) principle, the first phase of the task engaged teachers with the movie ratings data context. All teachers discussed specific contextual considerations of movie ratings, for example, the differences between the rating systems used by IMDb and Rotten Tomatoes and attempted to use contextual information to guide model decisions later in the task (cf. Pfannkuch, 2011). In the second phase of the task, using the *re-familiarise* (P2) principle, the teachers were first encouraged to extend familiar ideas of linear regression models to create prediction intervals without using code. The teachers demonstrated statistical thinking when they quantified the size of the prediction errors using an informal visual approach, using methods such as placing their pens on top of the computer screen.

Step 10 of the task aligns to the *describe* (P3), *match* (P4) and *adapt* (P5) principles and the teacher interactions with this step varied, perhaps because there was too much new information presented in one step. The code provided descriptions of the computational steps and output in terms of the prediction model, however, teachers struggled with matching or adapting at least one aspect of the code. In the first iteration of the design framework (Chapter 4), the task constructed used three separate steps that aligned to these design principles, which meant the teachers knew what computation was needed for each modelling step before reading the code and could focus on how the code syntax and structure tells the computer how to carry out each step (cf. Pruim et al., 2017).

The *adapt* (P5) principle appeared to be more successfully demonstrated in Steps 12 and 13. All pairs of teachers successfully integrated statistical and computational thinking when they adapted the code to source appropriate data from the API and modified the error value of their prediction model in response to the accuracy of the predictions. Using the *explore* (P6) principle, Step 14 encouraged teachers to explore changes to the code with respect to the data and models, with the expectation that new or unexpected outputs from these adaptions would stimulate an integration of statistical and computational thinking and new knowledge. Although

I have presented results that indicate statistical and computational thinking was observed for the teachers in earlier steps of the task, on reflection Step 14 could have asked the teachers to explore prediction models using a different combination of the variables and data available from the OMDb API.

5.6.1 Summary

I have provided some practical design solutions that balance the learning of new statistical and computational ideas. In particular, the use of *tinker questions* within a web-based task has the potential to develop learners' confidence with code experimentation. The web-based task, created using the *R* package *learnr*, provided easy access to new data to test prediction models and appeared to support the development of new statistical and computational ideas related to predictive modelling and APIs. Furthermore, for high school implementation, my web-based task has the advantage of only needing a browser to engage with the code-driven tool, rather than additional knowledge of computer systems and software installation. More research, however, is recommended on task design that supports teacher and student learning in data science at the high school level, which could include how teachers construct new tasks in line with the proposed design principles and considerations (cf. Sentance et al., 2019).

In Chapter 4, the randomisation test task (Task 3) was constructed to move learners from a GUI-driven tool to a code-driven tool for carrying out the randomisation test. In this chapter, I explored the introduction of code-driven tools through a predictive modelling task (Task 2) and demonstrated how the second iteration of my task design framework was modified and aligned to this task. In Chapter 6, I explore the introduction of code-driven tools through a classification modelling task (Task 1) and use the task and its implementation with teachers to evaluate the task design framework and examine integrated statistical and computational thinking.

Chapter 6: Classification modelling task

6.1 Introduction

The high school statistics curriculum needs modernising and expanding to include more of the data that students encounter in their everyday lives (e.g., Finzer, 2013; Gould, 2010; Ridgway, 2016). As students upload and share images through social media platforms and other digital communications, the use of digital images provides a relevant and important data context for teaching statistics. The analysis of digital image data can support understanding that data are numbers with context (Cobb & Moore, 1997) and can encourage students to integrate statistical and contextual knowledge, an essential aspect of statistical thinking (Wild & Pfannkuch, 1999). Furthermore, the digital technology context provides an opportunity to co-develop students' statistical and computational thinking, which aligns with the digital technology goals of New Zealand schools and recommendations for teaching data science (e.g., De Veaux et al., 2017; Gould, 2021).

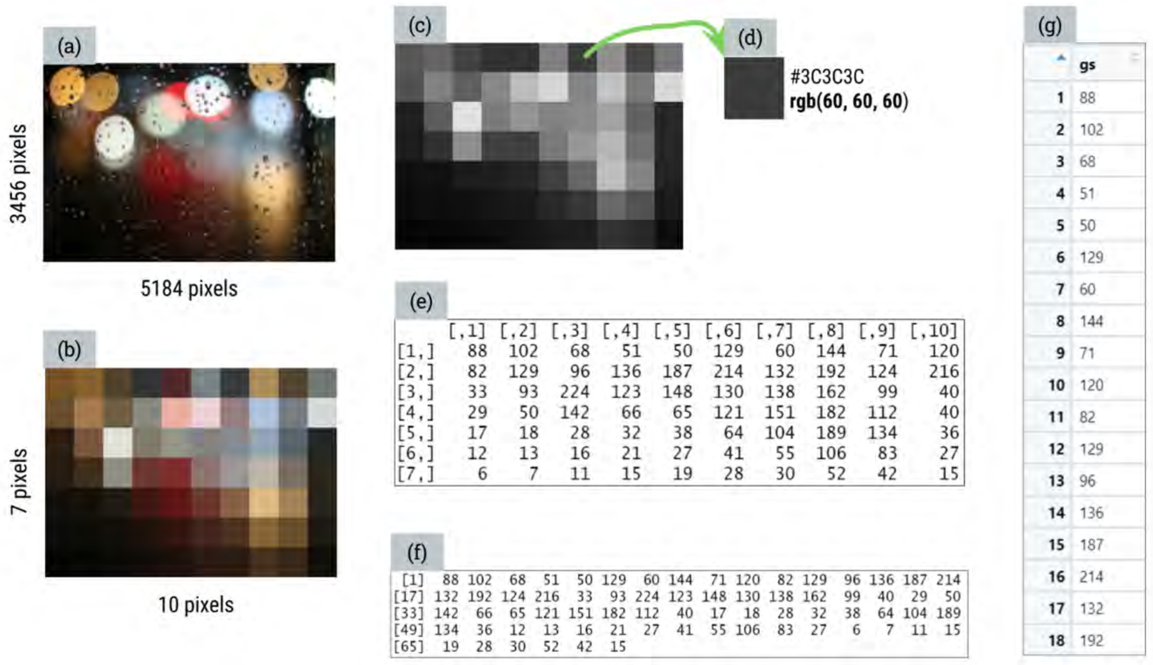
The opportunities for students to integrate statistical and computational thinking with digital image data can be broadened when introduced alongside new approaches to statistical modelling, such as the use of algorithmic models. Algorithmic models have been proposed as conceptually easier for students (e.g., Gould, 2017) and research with high school teachers suggests teachers with minimal knowledge of algorithmic models are able to quickly develop and interpret classification trees (Zieffler et al., 2021). Statistical modelling approaches using digital image data and classification trees, however, are not currently assessed by the national assessment system in New Zealand. To teach classification modelling with digital images at the high school level, statistics teachers will need access to tools and learning tasks for analysing digital image data.

Using a DBR approach, I created and refined a design framework to inform the development of new tasks to introduce code-driven tools through statistical modelling (Chapters 4 & 5). In this chapter, I explore the design and implementation of a task for introducing teachers to classification modelling with digital images (Task 1). I investigate if the task design framework could be applied to a task that uses a different source of data and a different statistical modelling situation, specifically digital image data and classification modelling. I also examine what statistical and computational ideas and thinking emerge when the task is implemented with teachers.

6.2 Teaching classification modelling with digital images

Images such as photographs provide an engaging and accessible modern data-context for teaching statistics, especially those shared on social media platforms such as Twitter or Instagram (e.g., Boehm & Hanlon, 2021; Fergusson & Bolton, 2018). Students can develop variables based on visual features of the photographs, by counting objects visible within the photograph or by sorting the photographs based on a specific quality (e.g., Bargagliotti et al., 2021, pp. 31—35; Fergusson & Wild, 2021). These kinds of activities begin to expand students' notions of data and provide encouragement to see opportunities for data creation everywhere. However, the analysis of *digital image data* involves more than just an awareness of data and the curiosity to learn from data. To learn from digital image data requires thinking that extends beyond integrating contextual and statistical knowledge (Wild & Pfannkuch, 1999) to include the computational (Gould, 2021).

Teaching computational image analysis can involve understanding digital representations of images; data structures that are different from the multivariate rectangular datasets most commonly used in statistics classrooms. High school statistics students are familiar with datasets where each row represents a different *case* or *entity,* and each column represents a different *variable* or *attribute* about that entity. Digital image data from a grayscale photo can be represented in this structure (Figure 6.1g), where each row represents a pixel from the image. However, this is not the initial data structure for a digital image, and when images are processed, they can also be reduced in dimensions. Figure 6.1 shows six different representations of the same photograph that capture the process of converting a colour photo to grayscale as well as reducing the dimensions of the image. The different representations illustrate some of the computational steps involved to create a data structure that can be used with classroom statistical software, as well as computational knowledge such as the RGB (red, green, blue) colour system and hexadecimal codes (six-digit combinations of numbers and letters defined by their mix of RGB).



```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    88  102   68   51   50  129   60  144   71   120
[2,]    82  129   96  136  187  214  132  192  124   216
[3,]    33   93  224  123  148  130  138  162   99    40
[4,]    29   50  142   66   65  121  151  182  112    40
[5,]    17   18   28   32   38   64  104  189  134    36
[6,]    12   13   16   21   27   41   55  106   83    27
[7,]     6    7   11   15   19   28   30   52   42    15
```

```
 [1]  88 102  68  51  50 129  60 144  71 120  82 129  96 136 187 214
[17] 132 192 124 216  33  93 224 123 148 130 138 162  99  40  29  50
[33] 142  66  65 121 151 182 112  40  17  18  28  32  38  64 104 189
[49] 134  36  12  13  16  21  27  41  55 106  83  27   6   7  11  15
[65]  19  28  30  52  42  15
```

	gs
1	88
2	102
3	68
4	51
5	50
6	129
7	60
8	144
9	71
10	120
11	82
12	129
13	96
14	136
15	187
16	214
17	132
18	192

Figure 6.1: Six different representations of a photograph: (a) colour photo; (b) dimensions reduced; (c) converted to grayscale; (d) HEX code and RGB values; (e) matrix; (f) vector; (g) data frame/table

The data structures associated with image data could provide genuine learning opportunities for mathematics, for example, the use of matrix representations (Figure 6.1e). However, using formal mathematics representations, notation, and formulae to introduce image analysis (see Li, 2018) could be a potential barrier for engaging a wide range of students. Therefore, when introducing the mathematics of digital image data structures, care needs to be taken to design learning tasks that highlight and promote key statistical concepts (Bargagliotti & Groth, 2016). For example, in the *Nanoroughness Task* discussed by Hjalmarson et al. (2011), students did not directly engage with the mathematical structure of the digital image data for grayscale photos. Instead, students were given physical grayscale photos where the levels of darkness (grayscale numeric values) were represented using a scale legend.

Opportunities to promote statistical reasoning without mathematical representations also exist when manipulating images, such as changing the contrast of a grayscale photograph using histogram equalisation. Exploring the distribution of grayscale values lends itself to reasoning with data distributions, where the data are the pixels of the digital image. Existing research about how learners reason with distributions (e.g., Bakker & Gravemeijer, 2004) can then inform task design, in particular reasoning about shape (e.g., Arnold & Pfannkuch, 2016), the role of context in interpreting distributional features, and learners' difficulties with interpreting histograms (e.g., Boels et al., 2019; Kaplan et al., 2014a).

It is also important to use digital image data within learning contexts that are genuine, and where the context is crucial to the design of the learning task (e.g., Weiland, 2016). Digital photographs are very commonly used to develop models that predict categorical or numeric outcomes, for example, predicting age from a photograph (see how-old.net), or classifying a photo as having either high or low aesthetic value (e.g., Datta et al., 2006). Decision trees are common algorithmic models used for classification, and have been included in high school data science or modernised statistics curriculum documents such as the International Data Science in Schools Project (IDSSP, idssp.org/pages/framework.html), Introduction to Data Science (IDS, idsucla.org), ProCivicStat (iase-web.org/islp/pcs) and ProDaBi (prodabi.de). Not only does the use of classification models with digital images provide a genuine learning context, but also classification problems are considered to be easier for learners to understand than regression problems (Gould, 2017).

Algorithmic models such as classification models (e.g., decision trees), however, are not developed in the same way as probabilistic models. Research involving high school statistics teachers indicates the need to consider the role of context and understanding of the use of training and validation phases (Zieffler et al., 2021). Teachers also need to be aware of the different sources of uncertainty in the modelling process, such as *objective* versus *subjective* uncertainty (Yang et al., 2019), and how these uncertainties might be articulated by students when completing learning tasks (Gafny & Ben-Zvi, 2021). Another consideration is for students to understand that algorithmic models are fallible just like human decision-making is, and therefore teaching and learning needs to account for the human dimension of data work (Lee et al., 2021).

There is also the question of what tools to use to teach classification models. I have proposed classifying tools for statistical modelling based on whether they are *unplugged*, *GUI-driven* or *code-driven* (Chapter 4). I describe *code-driven* tools as computational tools, which users interact with predominantly by entering and executing text commands (code), and *GUI*-driven tools as computational tools, which users interact with predominantly by pointing, clicking, or gesturing. It is a common approach within statistics education to use *unplugged* tools before moving to *GUI-driven* tools. Pedagogical approaches include using data cards to create visual representations (e.g., Arnold et al., 2011) or shuffling cards by hand to simulate random allocation of treatments to units (e.g., Budgett et al., 2013). Teaching materials from IDS and ProCivicStat include *unplugged* modelling with data cards before moving to the computer to develop classification models. Specifically, ProCivicStat uses the *GUI-driven* tool CODAP (Engel et al., 2019), whereas IDS uses a *code-driven* tool employing the programming language *R* (R Core Team, 2020).

Although *code-driven* tools can assist the analysis of digital image data, little is known about how teachers will balance learning new statistical and computational knowledge within the same task. Emerging research indicates that learners may frame problems as either statistical or computational when encountering issues with executing code (Thoma et al., 2018) and that building on familiar statistical ideas and matching to modelling actions may support teachers' introduction to code-driven tools (Chapter 4, Fergusson & Pfannkuch, 2021). There may also be benefits to using code with respect to lowering the cognitive demands of the statistical modelling task (e.g., Son et al., 2021), perhaps as code can be used to articulate modelling steps (e.g., Kaplan, 2007; Wickham, 2018). There is a lack of research involving high school statistics teachers' reasoning with digital image data, and none that I am aware of that involves them using digital image data to develop classification models.

6.3 Task design framework

Through retrospective analysis of two other tasks (Chapters 4 & 5), I explicated a design framework to introduce learners to code-driven tools through statistical modelling (Chapters 4 and 5). Because of the iterative nature of design-based research, I evaluated the second iteration of the design framework (Chapter 5) against the criterion of producing a more general design framework rather than one that was too specific to the nature of the tools and tasks used previously. My evaluation confirmed that the second iteration of the task design framework was also applicable to the classification modelling task.

6.4 Task design characteristics

I now discuss how the *second iteration* of my task design framework aligns to the task used in this chapter. The statistical modelling approach involved developing classification models using an informal method. The teachers were not introduced to any formal procedures for developing classification models in the task and instead were expected to reason visually with numeric distributions. Only the idea of a *decision rule* was introduced, and the task required teachers to use an aggregate measure of a numeric variable to classify cases as one of two levels of a categorical variable. The learning goal for the task was for the teachers to create a decision rule to classify grayscale photos as high contrast or no high contrast, based on the distributional features of the digital image data.

The design consideration of *the introduction of new knowledge* (C1) refers to the use, content and sequence of phases and steps within a task to introduce learners to new ideas. A summary of the three phases of the task and how they relate to the six design principles is provided in Table 6.1.

Table 6.1: Summary of task phases and design principles used

Phase	Summary of phase	Principle
1	Introduce digital image data	*Immerse* in data context (P1)
2	Introduce classification modelling ideas	*Re-familiarise* with statistical modelling ideas (P2) *Describe* computational steps of statistical modelling process (P3) *Match* statistical modelling steps to code chunks (P4) *Adapt* code chunks with slight modifications (P5)
3	Develop models to classify "high-contrast" grayscale photos	*Explore* "what if?" changes to code (P6)

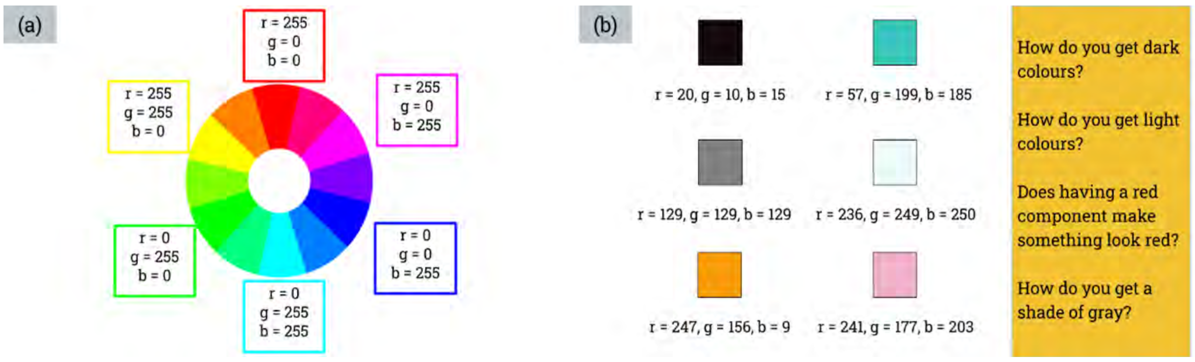
The design consideration of *the data used* (C2) refers to selecting and using different sets of data within the same data context for the task. The decision to use digital image data from grayscale photos provided data with different features that can be exploited to stimulate both statistical and computational thinking and was also aligned to the larger research study goal to provide a data science perspective for statistical modelling. The digital image data for the task were thirty popular photos of dogs sourced from the website *unsplash.com*. Each of these photos was reduced in size and converted to grayscale.

The design consideration of *the tools used* (C3) refers to combining different tools for statistical modelling (unplugged, code-driven, GUI-driven) and connecting actions or representations between tools within the task. Given that teachers had no experience using digital image data or classification models for statistical modelling, there were no familiar GUI-driven tools to use for this task. Consequently, there was a decision to use an *unplugged* approach to introduce classification modelling ideas. As the teachers were familiar with the combined dot and box plots produced by the software tool *iNZight* (Wild et al., 2021), a decision was made to use them. The design consideration of *the level of computational transparency* (C4) refers to how obvious the computations performed by the tool are to the learner. Hence, there was a decision to represent the data only using the combined dot and box plots to avoid introducing new ideas of data structures within the task.

I guided teachers through the task using presentation slides, live demonstrations, verbal instructions, and instructions embedded within interactive documents. The task was designed to take around 90 minutes to complete. Each phase of the task (see Appendix E) is now described and includes further explanations about how the design framework captures design decisions.

6.4.1 Phase one: Introduction to digital image data

The first phase of the task *immersed* (P1) teachers within a digital image data context so that they could meaningfully engage with the classification of grayscale photos later on. At the start of the phase, teachers were shown a colourful photo of jellybeans, given a brief explanation about digital images and pixels, and shown a colour wheel that demonstrated the RGB system for defining the colour of each pixel (Figure 6.2a). Teachers were then shown six different colours with their associated RGB values and worked in pairs to discuss answers to four questions that were designed to help teachers learn about the RGB system through noticing patterns (Figure 6.2b).



How do you get dark colours?

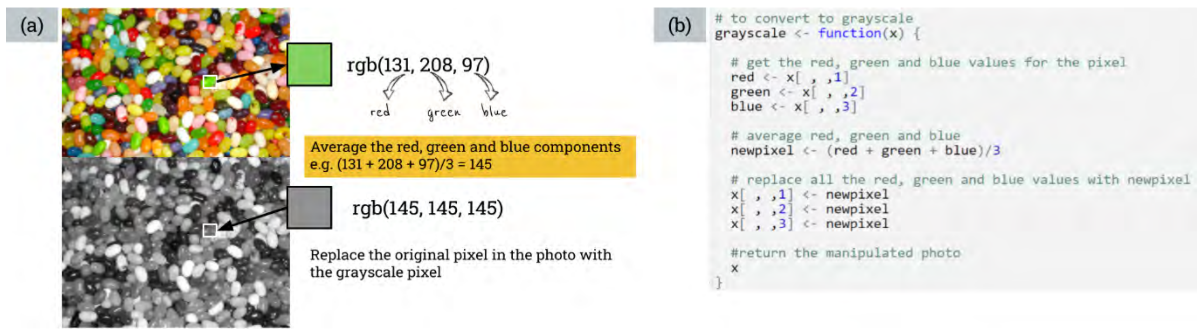
How do you get light colours?

Does having a red component make something look red?

How do you get a shade of gray?

Figure 6.2: Slides used to introduce the RGB colour system

I explained several methods for converting a colour photo to grayscale, one of which is to take the average of the RGB values for each pixel and replace the RGB values for that pixel with this one average value (Figure 6.3a).



Average the red, green and blue components
e.g. (131 + 208 + 97)/3 = 145

Replace the original pixel in the photo with the grayscale pixel

```r
# to convert to grayscale
grayscale <- function(x) {

  # get the red, green and blue values for the pixel
  red <- x[ , ,1]
  green <- x[ , ,2]
  blue <- x[ , ,3]

  # average red, green and blue
  newpixel <- (red + green + blue)/3

  # replace all the red, green and blue values with newpixel
  x[ , ,1] <- newpixel
  x[ , ,2] <- newpixel
  x[ , ,3] <- newpixel

  #return the manipulated photo
  x
}
```

Figure 6.3: Slides used to explain one process for converting a colour photo to grayscale

I then demonstrated using the programming language *R* to convert a digital image from colour to grayscale (Figure 6.3b), where the code was specifically written to connect to the visual explanation shown in Figure 6.3a. The teachers then watched me use the `grayscale` function to convert a colour photo of a cat to grayscale (Figure 6.4a and Figure 6.4b) and another function to create a plot based on a random sample of 50 pixels from the grayscale photo (Figure 6.4c).

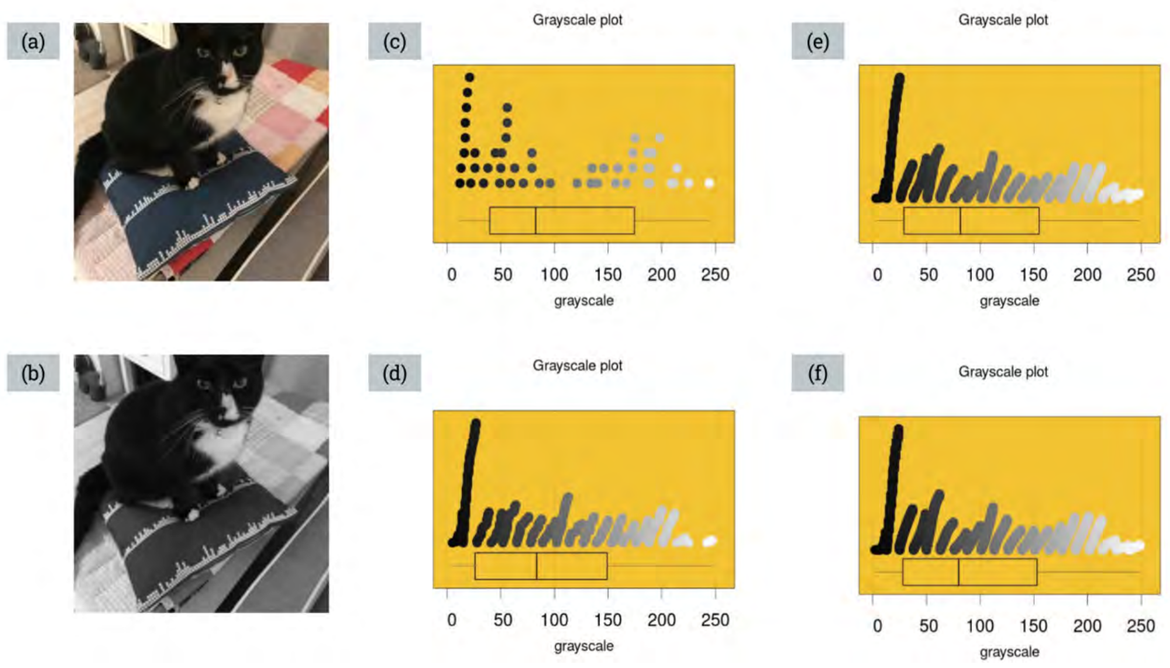


Figure 6.4: The colour and grayscale photos used for the demonstration and examples of the four grayscale distributions generated using increasing sample sizes

I explained that the plot contained a dot plot and a box plot, and that the shade of gray used for each dot was connected to the grayscale value of that pixel back in the photo. I then demonstrated the different plots created using a random sample of 500 pixels (Figure 6.4d), 5000 pixels (Figure 6.4e) and using all 91204 pixels (Figure 6.4f). The teachers were asked to discuss how the shape of the distribution changed and how long it took the computer to produce each plot as the sample size increased.

Phase one ended with each pair of teachers being given ten different grayscale photos and ten different plots constructed using a random sample of 500 pixels (Figure 6.5). Each pair of teachers received a unique set of ten photos. The ten photos that made up each set were balanced so that each group had a range of light, middle and dark photos, and contained both high and low contrast photos.

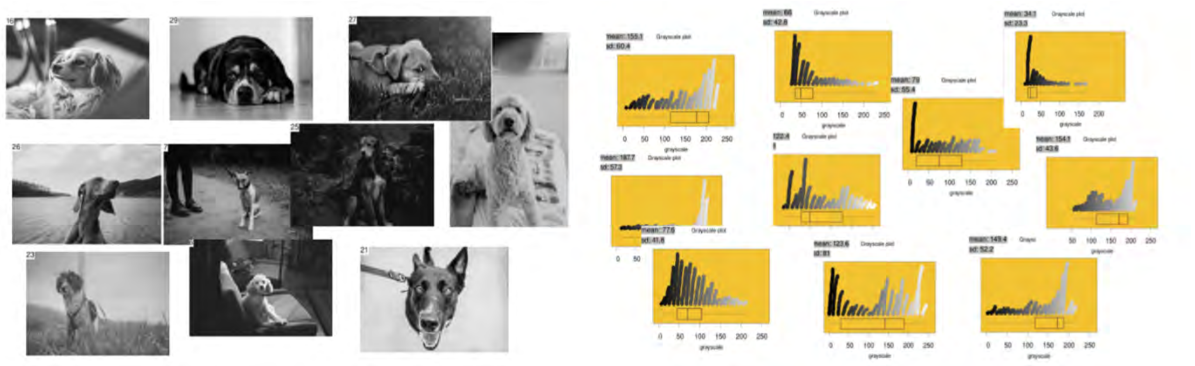


Figure 6.5: Ten different grayscale photos and ten different plots constructed using a random sample of 500 pixels

The teachers were asked to *connect* each photo with its dot plot, with the expectation that they would not be able to accurately connect each photo to its dot plot, a difficult task for photos with similar distributions of grayscale values. The purpose of the *connecting activity* was to stimulate discussion about the differing features of the sample distributions to lay foundations for reasoning with these distributional features later in the task.

6.4.2 Phase two: Introduction to classification modelling ideas

The second phase of the task introduced teachers to classification models and the idea of using an aggregate measure, such as the median grayscale value, to measure the overall “lightness” of a grayscale photo. The task design process for this phase considered how to *introduce new knowledge* (C1) such as decision rules and “training” and “testing” models, alongside more familiar ideas such as measures of central tendency and sample-to-population inference. With respect to *the tools used* (C3), a decision was made for teachers to use the code-driven tool at the end of the phase, because unplugged approaches were considered be more suitable for beginners to build familiarity with the computational steps for classification modelling.

As the teachers were unfamiliar with classification modelling, the *re-familiarise* (P2) design principle guided the decision to use familiar statistical ideas related to medians to introduce decision rules. Each pair of teachers initially worked with their own set of 10 photos and plots, where each photo was attached to a plot using a sticky dot. I asked the teachers to arrange the photos in order from darkest on the left to lightest on the right, and, after they had ordered the photos, I suggested that perhaps a computational method could be used to sort the photos from darkest to lightest. I asked the teachers to look at the median grayscale value displayed on the box plot and discuss whether the medians increased in size as the photos changed from dark to light. The direction to teachers to compare their visual judgements of lightness to a statistical measure based on the grayscale distribution was also designed to expose teachers to the difficulties of humans classifying high contrast photos as light or dark. An example of a conjectured "line up" is shown in Figure 6.6. Note that the median for each distribution of grayscale values (as indicated by the box plot) generally increases as the overall lightness of the photo increases. However, the photos and plots labelled 1 and 2 break the pattern of increasing medians.

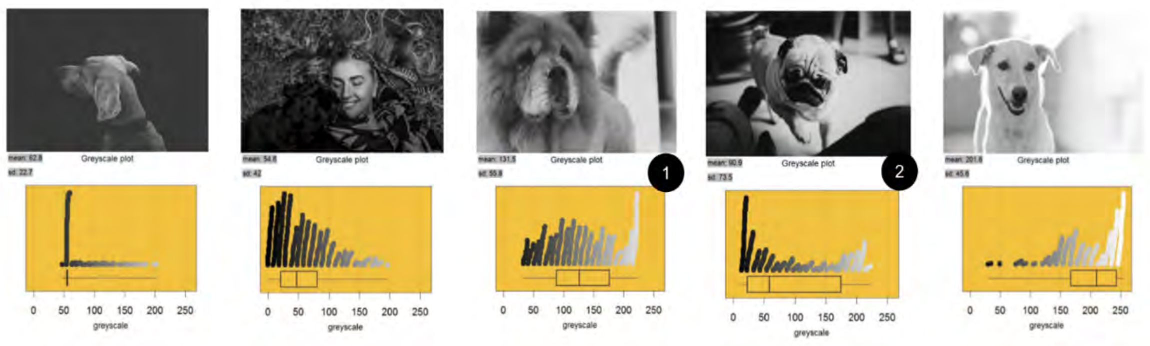
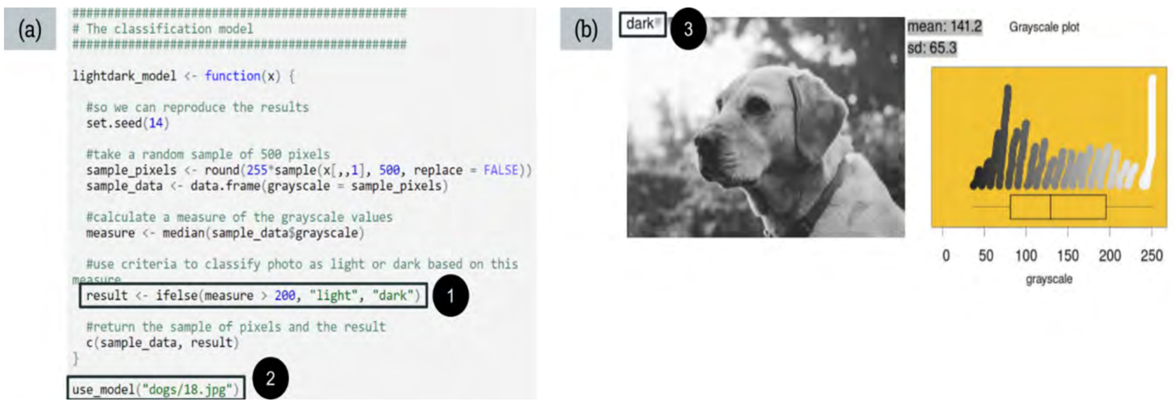


Figure 6.6: Example of a conjectured "line up" of photo-plot pairs

Continuing with the unplugged approach, the teachers were then asked to develop a model to sort the photos into light or dark based on the *median* grayscale value from a random sample of 500 pixels from the photo. I demonstrated that the computational steps involved reading in an image, taking a random sample of 500 grayscale values, and then deciding if the image was dark or light based on the median of the sample grayscale values being above or below a certain value. The teachers were asked to develop their own decision rule by first sorting the photos into "light" or "dark" visually, and then examining the *median* grayscale value for each sample distribution. The subjective labelling of the photos as "light" or "dark" by the teachers is not a typical part of learning about classification model, where it is more standard to provide students with data that has already been "correctly" labelled. However, similar to the reasoning provided by Podworny et al. (2021), in real applications of classification models, labels are assigned using subjective measures, and the teachers labelling the data themselves could help them understand the role of human decision making in the modelling process.

After the teachers decided on a “cut off” value, they had to count how many of their photos would be correctly classified according to their rule. I explained they had just “trained” their model and now they had to test it using new data. The decision to use a different set of dog images for “testing” their model was not just informed by the design consideration of *the data used* (C2) but because it was an important new idea for developing classification models that teachers had not used before. The teachers were instructed to swap their set of 10 photos and plots with another pair of teachers, to apply their classification model to this set of photos, and to count how many of the photos would be correctly classified according to their rule. The teachers were then given back their original set of 10 photos and plots and asked to *describe* (P3) their classification model to the other teachers and how well it worked on the “testing” data. Following this, the teachers were asked to discuss if they thought using the *mean* grayscale value might be a better way than the *median* to sort and classify grayscale photos in terms of their lightness.

I then demonstrated how to use *R* code to articulate their model based on their decision rule for the *median* to classify a particular photo as light or dark. Figure 6.7a shows the code provided and the two lines of code that were the focus of the demonstration, which are labelled 1 and 2. The line of code labelled 1 needed to be modified by the teachers to *match* (P4) their model, for example, using a different “cut off” number than 200 for the median grayscale. The line of code labelled 2 needed to be *adapted* (P5) by teachers to use the model with different dogs based on the photo number, for example, changing 18 to photo number 21. Figure 6.7b shows the physical photo-plot pairing recreated digitally for dog number 18, with the label “dark” added to the top left-hand corner of the photo, which is labelled 3.



```r
##################################################
# The classification model
##################################################

lightdark_model <- function(x) {

  #so we can reproduce the results
  set.seed(14)

  #take a random sample of 500 pixels
  sample_pixels <- round(255*sample(x[,,1], 500, replace = FALSE))
  sample_data <- data.frame(grayscale = sample_pixels)

  #calculate a measure of the grayscale values
  measure <- median(sample_data$grayscale)

  #use criteria to classify photo as light or dark based on this
measure
  result <- ifelse(measure > 200, "light", "dark")

  #return the sample of pixels and the result
  c(sample_data, result)
}
use_model("dogs/18.jpg")
```

Figure 6.7: Example of a classification model articulated with code with the output generated from the model

The teachers were then given access to a *RMarkdown* (Allaire et al., 2021) document that contained the *R* code shown in Figure 6.7a and were asked to *adapt* (P5) the code to *match* (P4) the classification model they had developed. At this stage of the task the teachers did not know if they had correctly connected their photos to the plots, so when they changed the code based on their photo numbers, they were able to check. The teachers were asked to *adapt* (P5) the provided code but change the classification model to be based on the *mean* grayscale value. The purpose of the light/dark classification task was to stimulate discussion about the grayscale distributions of high contrast photos and so motivate a need to develop a model to classify such photos.

6.4.3 Phase three: Exploration of "high-contrast" grayscale photos

The third phase of the task required teachers to develop their own rule for determining high contrast photos, using features of the sample grayscale distributions. Considering *computational transparency* (C4) and *the tools used* (C3), a decision was made to allow time for teachers to develop their classification model using an *unplugged* approach first, before providing them with a code-driven tool to *explore* (P6) changes to their model.

Teachers were shown a video featuring a photographer discussing high contrast photos (see youtube.com/watch?v=31qVHQbd0JU), for the dual purpose of explaining what high contrast was and also to reinforce that high contrast grayscale photos are a desirable aesthetic. The teachers were then asked to use their 10 photo-plot pairs from the earlier two phases to develop a model to sort the photos into high or low contrast. I encouraged them to look again at their photo-plot pairs, to split them into photos they thought were high contrast versus those they did not and to consider any features of the distributions of grayscale values that could be used to identify photos with high contrast.

After around five minutes of the teachers using only the physical photo-plot pairs to *explore* (P6) their ideas for classifying photos as high contrast, teachers were provided with a *RMarkdown* document that contained instructions for the rest of the phase and “starter code”, similar to the code shown in Figure 6.7a. At the start of the document, teachers were asked to write a couple of sentences describing how they developed their model in response to the following questions: *What did you notice about the photo/plot pairs? What feature of the grayscale pixels are you using? What is your criterion for when a photo is high contrast and when it is not?*

The teachers were then asked to adapt the code provided to articulate the classification model they had developed. The document also asked teachers to test their model using another random sample of 10 dog photos, with code provided that could be adapted and copied to support them to do this. Teachers were asked at the end of the document to write about what they learned from testing their model in response to the following questions: *How well did the model work with another sample? Can you identify any reasons why the model worked better/worse? Do you have any ideas of how to modify your model or approach?*

6.5 Analysis

The implementation of the classification modelling task took place during the first day of the professional development workshops. This was the first task that required teachers to use a code-driven tool and took place in the afternoon. In the morning session, teachers explored the popularity of cat and dog photos from the website *unsplash.com* (see Fergusson & Wild, 2021). The teacher pairings for this task were: Amelia and Ingrid, Alice and Naomi, and Harry and Nathan (pseudonyms have been used).

Drawing on existing statistical knowledge, all the teachers were able describe distributional features of digital image data. The teachers used these distributional features to create rules to classify grayscale photos, for example, as high contrast or not high contrast. In this case, the teachers attempted to connect different features of distributions, such as skewness or bimodality, to visual features of high contrast photos, using statistical measures such as means, medians and standard deviations. All the teachers successfully modified existing code to engage with digital image data and to articulate their classification model. I now present in more detail the results of the teachers' interactions with the phases of the task and the consequent thinking practices that arose.

6.5.1 Phase one: Introduction to digital image data

The focus for phase one was integrating new knowledge related to digital data with familiar statistical knowledge related to distributions. After the teachers were introduced to new ideas related to digital images, including converting colour photos to grayscale and visualising random samples of pixels from grayscale photos, they began to develop their thinking about digital data during the connecting activity used for this phase. Figure 6.8a shows Amelia and Ingrid's connected photos and plots.

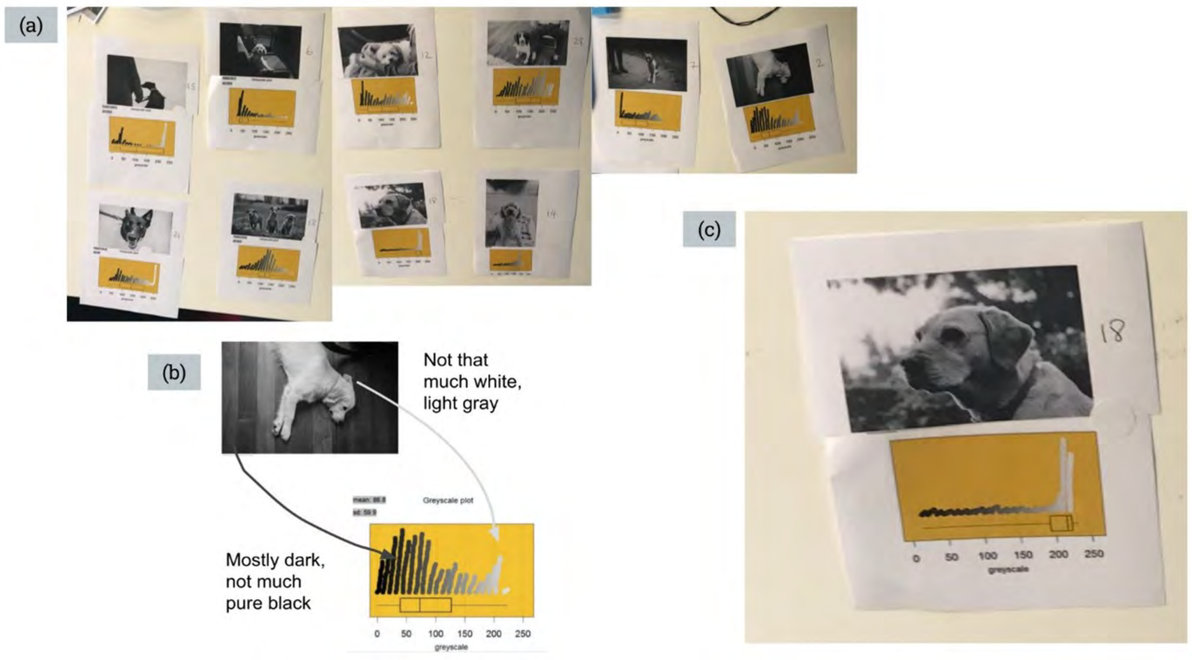

(a)
(c)
18
(b)
Not that much white, light gray
Mostly dark, not much pure black
Greyscale plot
0 50 100 150 200 250
greyscale
grayscale

Figure 6.8: Amelia and Ingrid's connected photos and plots

When asked to connect the grayscale photos to the plots of the samples of grayscale values, a *visual proportional* strategy was used by all pairs of teachers. Using this strategy, the teachers attempted to estimate the proportions of different shades of gray for each photo and then tried to connect these proportions to the dot plot representation of the sample of grayscale values. This was evident through the teachers' use of descriptions such as "more white" and "a lot of black." The teachers did not use numeric values for the shades of gray, which could have been read from the axis of the plots. Instead, they used descriptions of shades of gray such as "pure black" and focused on distinctive areas of darker shades of gray and lighter shades of gray (Figure 6.8b).

When describing distributions *during* this connecting activity, examples of words used included "extreme ends" or "edges" when referring to tails and "high pitched" or "spikes" when referring to modal grayscale values. These words, and the *visual proportional* strategy, indicated that the teachers were focused on *distributional shape*. More formal descriptions of distributional features, such as "skewness", "bi-modality" or "outliers", were used near the end of the activity when teachers compared their photo-plot connections with each other and within a researcher-led group discussion. For instance, the following discussion between three of the teachers was initiated by me asking if they had used the summary statistics printed on the plots to help them connect plots with photos.

> Naomi: No, it was the distribution, it was totally about the distribution.
>
> Amelia: We looked at things like bimodal, lots of contrast, or if it was particularly at one end if it was dark or particularly at the other one end if it was light.
>
> Naomi: Also, how far up the scale it was, wasn't just the shape. Was it pure white, or the grayer shades?
>
> Nathan: We were focusing on dots, focusing on the grayscale of the dots.
>
> Naomi: But, actually, variation was very important, because it wasn't just the mean it was also, "Is it bimodal?", as opposed to a normal [shape].

Note in this discussion that from the teachers' own perspective, *distributional shape* was the main factor used to make matches. Naomi also referred to "how far up the scale it was .... was it pure white, or the grayer shades?" and Nathan referred to "... focusing on the grayscale of the dots", both confirming the use of a *visual proportional* strategy.

The teachers did struggle, as anticipated, to match some photos using a *visual proportional* strategy. One contributing factor to their difficulties was that the grayscale plots were based on a random sample of 500 pixels. Towards the end of the activity, after the teachers had walked around the room and looked at the other pairs' matches, Naomi and Amelia discussed why connecting photos to plots using the *visual proportional* strategy may have been an issue. The discussion began when Naomi noticed how Amelia and Ingrid had matched photo number 18 to a dot plot (Figure 6.8c).

> Naomi: I would not have picked that one ... I'm just thinking there isn't enough white to be down there ... it's a little bit of white but it's not the biggest.

> Amelia: Because it’s just a sample of the pixels, you could have got a sample … so you might not have got as many from here [pointing to an area on the photo] as you would expect.
>
> Naomi: But a sample of 500 should be representative.
>
> Amelia: You’re right, but if you’ve got just an area, like this one where that’s the area where all the white is, if that area wasn’t sampled quite as much as that area [pointing to an area of black], you wouldn’t get the same kind of picture [referring to the dot plot] … but you’re right, 500 is a pretty big sample.

I interpreted that Amelia believed a random sample of 500 pixels *may* provide insight into the underlying shape of the distribution of grayscale values for all pixels, but that she also was aware that the proportion for each of the possible 256 individual shades of gray could vary considerably between samples. However, Naomi appeared to believe that for any distribution, a sample of 500 should be representative, perhaps reflecting a lack of understanding that distributional shape is dependent on both the sample size and space (the number of possible outcomes).

Another strategy used to connect photos with plots was demonstrated by Nathan and Harry, who attempted to order their photos from darkest to lightest using a *visual proportional* strategy, before connecting the plots to the photos. Amelia and Ingrid stated in the group discussion at the end of this phase that they had seen this strategy being used and tried to use it but, “the ones we had left were all the gray ones, it was really hard to put them darkest to lightest, using the mean wouldn’t have even been helpful even if we had used it.” Indeed, the *visual proportional* strategy appeared to be the most successful for teachers when the visual features of the photos were “strong”, with Amelia commenting that, “where it [the photo] had contrast, you’re looking for bimodal. Where it was really dark or really light, you were looking for skewness.”

The results from phase one indicated that teachers had begun to develop new ways of thinking about distributions. They were able to conceive of shades of gray from a photo as numeric data points and to use visual proportional and distributional reasoning to make connections between features of plots and photos. Connecting grayscale photos with features such as high contrast using *human* visual judgements was easier for the teachers than connecting photos with less distinctive features, and these judgements were also influenced by understandings related to sampling variation. Overall, the thinking demonstrated in this phase was mainly statistical and was focused on making sense of and describing features of the distributions of grayscale values.

6.5.2 Phase two: Introduction to classification modelling ideas

The focus for phase two was familiarising teachers with classification modelling ideas through drawing on familiar statistical ideas related to medians and means. When asked to order the photos from dark to light, all pairs of teachers found photos where they struggled to decide their "lightness" position relative to the other photos. Ingrid explained that, "… these [photos] are the hard ones because of the high contrast" and Naomi elaborated, "high contrast makes us think of light … because when we see high contrast, we've got a lot of light around." All pairs of teachers expressed a lack of confidence that they had ordered the photos "correctly" from darkest to lightest.

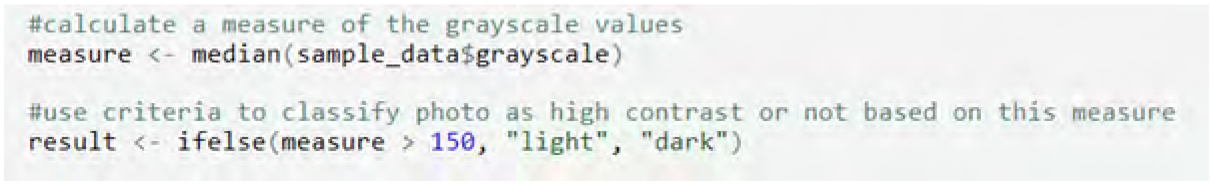
I asked the teachers to look at the medians of distributions attached to the sorted photos and to examine if the medians increased in size as the photos increased in lightness. The teachers observed that for their set of ten photos and plots, the medians did not always increase, which led them to perceive that they were at fault and that they had incorrectly ordered the photos from darkest to lightest. This perception was captured in a discussion between some of the teachers and me below, which took place after the teachers had developed their own decision rules for classifying a photo as "dark" or "light" based on the median grayscale value.

> Amelia: We struggled to say which was light and which was dark, I don't know that a human is very good at this.
>
> Naomi: No, that's right, I would totally agree!
>
> Amelia: If you're training the computer to do it, you might be training the computer to do it better than you could do it, so you're not like trying to get it to match the human model, it's [the computer] doing it better.
>
> Researcher: It depends, your target audience is still how a human perceives lightness or darkness.
>
> Amelia: Yeah, but to excuse the pun, there's a lot of gray in the middle.
>
> Researcher: Yeah, shades of gray!
>
> Naomi: As humans we're responding to other things than the median. I noticed that if something was taken in lower light, I would want to classify it as dark rather than just looking at the amount of light and dark because my brain starts thinking, "Oh

yes, darker picture, taken in lower light" even if maybe there's quite a lot of whiteness in it.

Researcher: So, what we're doing at this stage is we're just developing an idea. We're not saying the median is the best way to do it or that it even is possible. It's this idea of "What if?" Could we try this out, could we try and classify photos using the median?

Note that Naomi and Amelia articulated why human decision making might not be as consistent as the median measure used by a computer. As the goal of the classification model was to reproduce human judgements using the median, I reminded them the model is based on human perception and that the model may not be correct. Some evidence that the teachers were able to incorporate subjective human aesthetics as the basis for the classification model was the "cut off" value used for the decision rule. Figure 6.9 shows the decision rule used by Harry and Nathan, articulated with *R* code.



```
#calculate a measure of the grayscale values
measure <- median(sample_data$grayscale)

#use criteria to classify photo as high contrast or not based on this measure
result <- ifelse(measure > 150, "light", "dark")
```

Figure 6.9: The decision rule used by Harry and Nathan

The code shows that photos with a sample median grayscale value greater than 150 are classified as "light." Harry explained that they, "… found the cut off point for the median was higher than the middle of the distribution, as to our eyes it was more natural to put in the median where we did." The teachers also demonstrated some understanding of difficulties with dichotomising numeric variables when they tested their classification model on ten photos from another pair of teachers and identified that photos with median grayscale values "around the boundary" were often misclassified using their decision rule.

Although the focus for this phase was on introducing new computational ideas related to classification modelling, the teachers also continued to develop new ways of thinking about distributions and the impact of sampling variation. This was apparent when the teachers were asked to consider whether the median or the mean would be a better measure for the overall lightness for a grayscale photo. Amelia shared her reasoning in the following excerpt:

> The reason we thought the median was better was that where you've got pictures with like a big chunk of dark or a big chunk of light, you often have got the skewed distribution, so it would pull the mean one way. But that big chunk of light makes the picture look light, so you actually want to go with the median because you want to go with that big chunk.

The other teachers shared similar reasoning about a preference for using the median as the measure of overall lightness that best matched a *human visual judgement*. None of the teachers commented on the usefulness of using a measure of central tendency for distributions that were bimodal, however, Nathan indicated that he had begun to consider measures of spread when he agreed with using the median, "… because of the variation that you have here with your lights and darks." The impact of sampling variation on the performance of the classification model was also discussed by Ingrid, when she observed that the median jumped around more for random samples of 500 pixels than the mean did.

After the teachers had developed their decision rule to classify photos as "dark" or "light" based on the median grayscale, I realised that more information about classification models was needed. In particular, I reminded the teachers that for a classification model, "the goal is not 100% correct" and that they needed to be careful not to overfit their model. This new knowledge was then used by the teachers as part of their evaluation of the classification model for this phase. To illustrate, Harry reminded Nathan that, "… there is no perfect model, 90% will do" and not to focus too much on "getting the model to work" for special cases. Ingrid also specifically discussed being mindful of "overfitting" when describing their model to the other teachers at the end of the phase.

The results from phase two indicated that teachers had begun to develop understanding of how a human decision, such as subjectively measuring the lightness of a grayscale photo, could be automated using a classification model based on statistical properties of data. The teachers also appeared to recognise that classification models are evaluated based on the percentage of correct classifications and that this percentage may not be 100%. The use of familiar statistical ideas within the context of digital image data seemed to encourage new ways of thinking about the use of the median and mean to summarise a distribution. Hence, the teachers appeared to be connecting statistical and computational ideas in their thinking.

6.5.3 Phase three: Exploration of "high-contrast" grayscale photos

The learning goal for phase three was that teachers would integrate statistical and computational ideas as they developed and used models to classify "high contrast" grayscale photos with *R* code. This phase was less structured than the previous phases and provided an opportunity to observe how the teachers applied the new statistical and computational ideas introduced earlier. To gain an understanding of each pair's modelling process, the transcripts and screen recordings were analysed with respect to how much time in minutes each pair of teachers spent developing their model, articulating their model using code, and checking or changing their model. The analysis also considered when the unplugged (physical) photos and plots were used and when the code-driven tool was used. A visual summary of this analysis is shown in Figure 6.10.

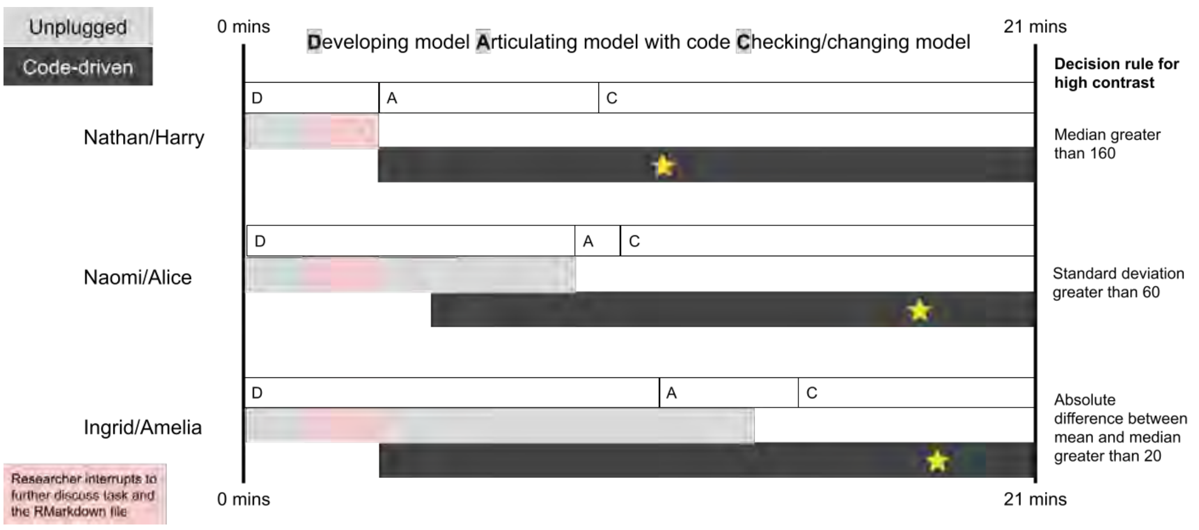


Figure 6.10: A visual comparison of the modelling process used by each teacher pair

Figure 6.10 also describes the classification model developed by each pair of teachers, for example, Ingrid and Amelia's final decision rule for high contrast was an absolute difference between the mean and median greater than 20. The star indicates the time at which the pair of teachers expressed that they were happy with their model. I now use Figure 6.10 and additional results to compare and describe the modelling processes used by the three pairs of teachers.

The process used by all the teachers to *develop* a classification model for high contrast grayscale photos involved:

1. making a conjecture about why the photo is high contrast, using human visual analysis/observation

2. making a conjecture about what statistical measure(s) might capture what the human has analysed/observed, using the sample distribution of grayscale values for that photo
3. formulating a decision rule using a statistical measure from the sample of grayscale values and using this rule with at least one photo to see if it "worked."

Figure 6.10 shows each pair of teachers spent differing amounts of time *developing* their model, ranging from around two minutes for Nathan and Harry to around 11 minutes for Ingrid and Amelia. The use of the tools provided also differed for each pair of teachers. As Figure 6.10 illustrates, Nathan and Harry did not use the unplugged tool (physical photos and plots) and the code-driven tool simultaneously while *developing* their model, in contrast to the other pairs of teachers. Notably, Ingrid and Amelia used both tools simultaneously for around 10 minutes while *developing* their model and then *articulating* their model with code. All pairs of teachers successfully *articulated* their model using *R* code, with the amount of time for each teacher pair depending on the complexity of their decision rule and their familiarity with using the code-driven tool. Naomi and Alice spent the longest time *checking* or *changing* their model before expressing happiness with their model. When *checking* or *changing* their model, the teachers did not always clearly differentiate between using training data or testing data.

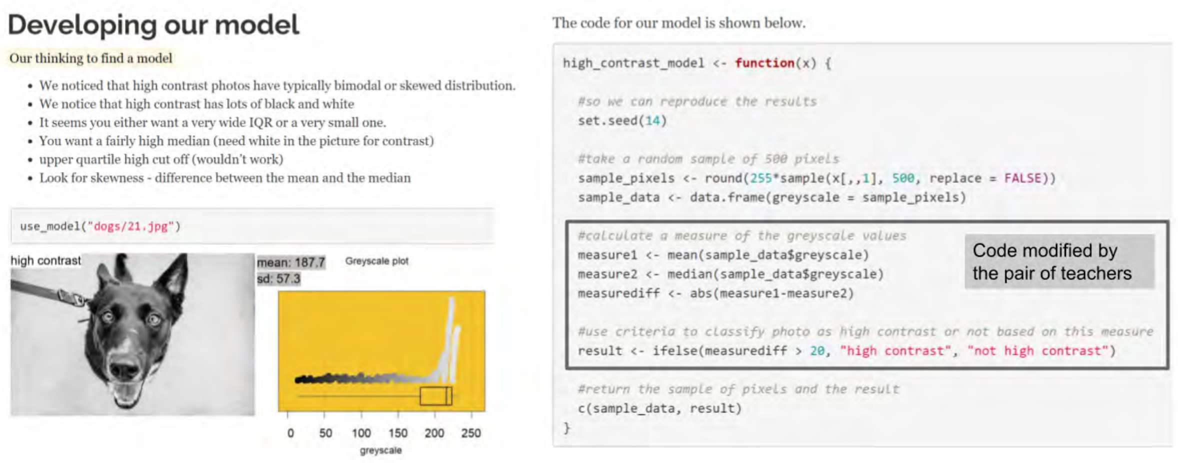
I now examine each pair separately to identify thinking practices that emerged as they developed models to classify "high contrast" grayscale photos. After doing a Google search for "high contrast", Nathan and Harry discussed high contrast in terms of bimodality but then used the same model they developed for classifying "light" photos to classify the physical grayscale photos as "high contrast." After the researcher interrupted the teachers to discuss how to use the *RMarkdown* document provided for this phase, Nathan and Harry moved to the computer and only used the code-driven tool for the rest of the phase. They took much longer to articulate their model than the other teachers, struggling at first to figure out how to run the code and consequently visualise the results within a *RMarkdown* document.

Nathan and Harry were much quicker to accept their model, with Nathan stating, "we accomplished our task!" after testing just one photo. When the researcher asked how many photos they had tested, Nathan replied, "one so far … before we continued, we wanted to make sure it worked." I interpret Nathan's reference to making "sure it worked" to be a reference to their code working, in that *computationally* they were able to make their model work. The computational focus appeared to be confirmed later in the phase when Naomi and Alice shared the model they had developed, which was based on standard deviation, and Nathan said, "we probably should have changed to standard deviation or something ... we just went into robot mode!” Standard deviation was a feature Nathan and Harry had discussed with reference to high contrast in phase one, and therefore after being given the model from Naomi and Alice, they were able to adjust their code and use Naomi and Alice’s model to classify a few photos as high or low contrast using the standard deviation.

Naomi and Alice began the phase with a pre-determined idea to use spread as the statistical measure to identify high contrast photos, as they had noticed in phase two that high contrast photos tended to have grayscale distributions with large standard deviations. Naomi stated, “variation is what was important”, and consequently they explored the interquartile range and the standard deviation as measures for the decision rule of their classification model. When they sorted their photos into high and low contrast, they did not always agree on whether an individual photo was high contrast or not. When disagreements arose, these were often resolved by sorting the photos according to their current decision rule and then considering if they still believed that the photo was high contrast or not.

Naomi and Alice used all ten of their photos to develop their model, and after quickly articulating their model with code, checked their model with “new” photos from outside their training data. The teachers agreed on a decision rule for their classification model that photos with a “standard deviation greater than 60” would be classified as high contrast. Like phase two, when the teachers demonstrated understanding of the difficulty of dichotomising numeric variables, Naomi and Alice realised that photos with “standard deviations close to 60” were “borderline” and the most likely to be misclassified.

Ingrid and Amelia demonstrated a more exploratory approach and consequently took longer to develop their model. Figure 6.11 shows screenshots from the teachers’ *RMarkdown*-generated HTML document and provides a description of the thinking they used to develop their classification model and the code they modified to articulate their model.


Developing our model

Our thinking to find a model

- We noticed that high contrast photos have typically bimodal or skewed distribution.
- We notice that high contrast has lots of black and white
- It seems you either want a very wide IQR or a very small one.
- You want a fairly high median (need white in the picture for contrast)
- upper quartile high cut off (wouldn't work)
- Look for skewness - difference between the mean and the median

```
use_model("dogs/21.jpg")
```

high contrast

mean: 187.7
sd: 57.3

The code for our model is shown below.

```
high_contrast_model <- function(x) {

  #so we can reproduce the results
  set.seed(14)

  #take a random sample of 500 pixels
  sample_pixels <- round(255*sample(x[,,1], 500, replace = FALSE))
  sample_data <- data.frame(greyscale = sample_pixels)

  #calculate a measure of the greyscale values
  measure1 <- mean(sample_data$greyscale)
  measure2 <- median(sample_data$greyscale)
  measurediff <- abs(measure1-measure2)

  #use criteria to classify photo as high contrast or not based on this measure
  result <- ifelse(measurediff > 20, "high contrast", "not high contrast")

  #return the sample of pixels and the result
  c(sample_data, result)
}
```

Code modified by the pair of teachers

Figure 6.11: The classification model developed by Ingrid and Alice

Ingrid and Amelia appeared to be more open to exploring more than one way of classifying high contrast photos and considered bimodality, skewness, and large proportions of "black" and "white" as features of the grayscale distributions that might indicate a high contrast photo. The teachers attempted to translate these distributional features into different decision rules, for example, by noticing high contrast photos have either "a very wide IQR or a small one" or that "you want a fairly high median." Ingrid and Amelia continued to use the unplugged *physical* photos and plots alongside the code-driven tool when developing their model. Using both tools appeared to help them quickly repeat the process of making conjectures about high contrast based on human visual analysis of a photo, making conjectures about what statistical measure(s) might capture the human visual analysis, and then formulating and using a decision rule based on a statistical measure to see if the rule worked.

Ingrid and Amelia also considered how their conjectures for statistical measures and decision rules could be expressed using code. For example, after examining the physical plots in front of her, Ingrid remarked, "you could look at the min and the max ... I'm thinking about things you could actually put into our model." Note in Figure 6.11, the teachers described that they wanted to use a decision rule based on the upper quartile but it "wouldn't work." Indeed, Ingrid and Amelia tried to modify the code to create a measure based on the upper quartile, but as this was not a specific function provided by the code-driven tool, they disbanded this attempt. Similarly, Ingrid stated, "it's kind of hard to find a model that will pick up both skewness and bimodality at the same time", referring to their knowledge about how to articulate this model computationally with the code provided, for example, by using more than one "if/else" statement.

Another reason why the development of their model took longer than the other teachers was that for each idea they had for a decision rule, they would try it out with several photos. If they found one photo that wasn't correctly classified by the rule, after confirming that they still thought it was high contrast, they would discard the rule. Like Naomi and Alice, they were prepared to change their judgement of whether a photo was high contrast or not, to get their model "to work." However, towards the end of the development phase of their model, Ingrid suggested that they should forget about one of the photos they were using to develop a model, saying, "maybe it's just an odd ball." Although Amelia initially resisted, she then agreed with Ingrid's justification that they are, "trying to find a model that will work in many cases but not necessarily perfect." I interpreted this change of approach as an indication that the teachers had begun to evaluate a classification model from the perspective of *Does our model get the photos correct most of the time?* rather than *Can our model cope with the "weird" or "tricky" photos?*

The results from phase three indicated that all the teachers had begun to understand how to develop and compare classification models. They appeared to understand that they could create new measures, which took different forms of variation into account, and they demonstrated they could articulate their decision rule with code. By drawing on both statistical and computational ideas in their modelling process, the teachers seemed to demonstrate integrated statistical and computational thinking. The last phase of the task was described by Naomi, with agreement from the other teachers, as providing an important learning experience of "trying to take something complex and create a statistical measure for it."

6.6 Reflection

One objective of my research was to identify statistical and computational thinking practices that might emerge when teachers were exposed to a new learning environment involving digital image data and classification modelling. The design decisions made when constructing the task seemed to provide a positive "first exposure" to classification modelling with digital image data.

Overall, the teachers appeared to extend their statistical thinking within the context of digital image data, and the tasks appeared to support them to integrate statistical and computational thinking. According to Lee et al. (2011), computational thinking involves abstraction, automation, and analysis. The teachers appeared to show these characteristics while exploring how to classify "high-contrast" grayscale photos. I observed that all teachers were able to: connect visual features of grayscale photos with features of sample distributions of grayscale values (analysis); create rules to classify grayscale photos in different ways (abstraction); and use code-driven tools to interact with digital image data and to articulate classification models (automation).

I now discuss some specific design decisions: framing the task in terms of distributions; using a *data science unplugged* approach; and encouraging human-driven informal model building. I make tentative links between these decisions and the results presented about teachers' thinking practices and then I reflect on my task design framework.

Framing the task in terms of distributions

The task directed teachers to reason with distributions throughout the task rather than leaving the analytical approach open to the teachers (cf. Hjalmarson et al., 2011). I propose this decision provided a way for the teachers to "look" at the digital image data through a familiar lens (cf. Wild, 2006) and supported the introduction of new computational knowledge by extending the familiar into unfamiliar data science approaches (Biehler & Schulte, 2017). A key task design feature appeared to be the *connecting activity*, where teachers physically connected photos (representations of the population distributions of grayscale values) with dot plots (representations of sample distributions of grayscale values). The activity stimulated teacher discussion about *distributional shape* (cf. Arnold & Pfannkuch, 2016) and helped to support the teachers to successfully reason with *hierarchal* data structures inherent in digital image data (Figure 6.1) through initially considering the pixels as the *cases* belonging to each photo. Once the teachers were able to connect the sample distributions of grayscale values with the grayscale photos, the connected photo-plots became the cases that could be summarised using a measure such as the median. The approach to draw teachers' attention to the median grayscale of the distribution and the lightness of the connected grayscale photo is consistent with the findings of Arnold et al. (2011).

Using a data science unplugged approach

The decision to focus on distributions for this task was strongly linked to the decision to use a *data science unplugged* approach. The use of physical grayscale photos and dot plots of sample grayscale distributions removed the need to introduce an unfamiliar GUI-driven tool approach for analysing digital image data and reduced the focus on mathematical structures. The physical sorting of the connected photo-plots into different groups for classification (e.g., light versus dark) appeared to offer similar benefits to the "hands-on" activities used within learning progressions for simulation-based inference with respect to modelling ideas (e.g., Chance et al., 2004; Zhang et al., 2021). I also observed that some teachers continued to refer to physical stimuli even when they had access to the code-driven tool, similar to what was found with the randomisation test task (see Chapter 4).

The *data science unplugged* approach used for the task is also consistent with the pedagogy described by Shoop et al. (2016) with respect to teaching robotics, where students work together to build models for computational solutions and present these models to the class for discussion before developing code. My findings appeared to demonstrate that teachers were able to learn new ideas related to classification models and describe the computational steps in their own words, before articulating their model using *readable* code (Wickham, 2018). The readable code was made possible through functions that were named to match physical and described actions.

However, a limitation of the *data science unplugged* approach is that the teachers' experiences with evaluating classification models were *small scale*. The training and testing data sets only contained 10 photos each and the hands-on approach prevented "scaling up" the evaluation of their classification models. Similar to the teachers observed by Zieffler et al. (2021), the approach did promote some initial modelling approaches that were based on overfitting specific features of grayscale photos. As a "first exposure" learning task, the task seemed to provide a foundation for further development of classification modelling ideas.

Encouraging human-driven informal model building

The decisions to frame the task in terms of distributions and to use a *data science* unplugged approach were connected to the decision to encourage human-driven model building. Heeding the same call as Lee et al. (2021) to take a humanistic stance towards data science education at the school level, the task provided students with personal and direct experiences with data and measurement. Notably, the sample distributions of grayscale values were provided for the teachers, and no pre-labelled data sets were made available, as is commonly the case with introductory classification modelling activities (e.g., Engel et al., 2019; Zieffler et al., 2021). Similar to the task developed by Horton et al. (2022) to explore how learners produce data from text to classify clickbait, the teachers needed to connect features they perceived as humans (e.g., high contrast grayscale photos) to features of the data (e.g., skewness of distributions) to computer extractable features (e.g., calculating the difference between the mean and median for a random sample of grayscale pixels), before they could develop rules that could be used to classify photos (Figure 6.11).

These human-driven decisions led to uncertainty with the modelling process, particularly as the teachers often doubted their own ability to classify photos as light or dark, or as high or low contrast. In my findings, I presented examples of the teachers grappling with the differences between how humans and computers make decisions. I contend that by not providing a complete and accurate data set and relying on human choices for both the measures and decision rules, the different sources of uncertainty that are faced by data analysts (cf. Yang et al., 2019) were effectively incorporated into the learning task (cf. Podworny et al., 2021).

Even though formal classification models were not introduced, I note that when Datta et al. (2006) attempted to create classification models for aesthetics of photographs using formal computational approaches, they discussed similar difficulties that my teachers discovered, specifically issues with dichotomising numeric variables. A teaching challenge is how to combine *thinking like a computer* and *thinking like a human* (cf. Biehler & Fleisher, 2021). On the one hand, statistical thinking requires learners to understand that data are numbers with context (Cobb & Moore, 1997) and thus humanistic perspectives of model outputs are needed that account for contextual implications.

On the other hand, I found that the context did at times distract the teachers from forming more general ideas about statistical models (cf. Biehler & Fleisher, 2021; Pfannkuch, 2011; Zieffler et al., 2021). Similar to Hjalmarson et al. (2011), I found that not all teachers developed a statistical measure that incorporated the variation of grayscale pixels within a photo. If human decisions are to be encouraged as part of modelling approaches, consideration may be needed of graphicacy theory, which states that what learners see in the data depends on their prior knowledge and experiences (Friel et al., 2001).

Task design framework

This chapter reports on the third task pertaining to my research, the purpose of which is to propose a design framework to construct and implement tasks for introducing code-driven tools through statistical modelling. Across the three tasks, my design framework seems to be effective in explicating design decisions that introduced the teachers to statistical modelling using code-driven tools: (1) the *randomisation test* using code-driven tools, moving from familiar GUI-driven tools (Chapter 4); (2) *predictive modelling* using code-driven tools (Chapter 5); and (3) *classification modelling* using unplugged and code-driven tools (this chapter).

The design principles and considerations were based on teaching strategies sourced from statistics education and computer science education research (e.g., Garfield et al., 2012; Sentance & Csizmadia, 2017). Additionally, the design principles and considerations of my design framework are similar to those of emergent frameworks within statistics education and computer science education. For example, the *practicing connections* framework (Fries et al., 2021) explicates that learning tasks for statistical modelling should help students make and practice connections between concepts and different representations. The three tasks seemed to support the teachers to make connections between different representations of data, computational actions, and models. The *PRIMM* framework (Sentance et al., 2019) explicates a structured approach to developing a learning task for computer programming using the learning actions *Predict*, *Run*, *Investigate*, *Modify*, and *Make.* My framework recommends a similar structured approach using design principles (see Table 6.1), the design consideration of the *introduction of new knowledge* (C1), and a similar learning goal for learners to *explore* (P6) and thus create new ideas.

I contend that my framework provides further guidance than the PRIMM framework for the development of statistical modelling tasks that introduce code-driven tools in at least two notable ways. The design principles of *immerse* (P1) and *re-familiarise* (P2) encourage the design of learning tasks where data contexts and statistical modelling ideas are developed initially *without* using code-driven tools. I propose that these design principles and the design consideration of the *data used* (C2) support statistical thinking through the development of learning activities that encourage the integration statistical and contextual knowledge (Wild & Pfannkuch, 1999). Furthermore, in line with Biehler (2018), my framework considers the relationship that exists between tools, task design and learners' statistical and computational conceptions. The design principles *describe* (P3), *match* (P4), *adapt* (P5), and the design considerations of the *tools used* (C3) and the *level of computational transparency* (C4), explicate *how* to make connections between statistical modelling experiences, actions and representations for code-driven tools and unplugged or GUI-driven tools.

6.6.1 Summary

I have proposed some practical design solutions for balancing the learning of new statistical and computational ideas when introducing code-driven tools for statistical modelling. Using an *unplugged data science* approach, the task provided an accessible introduction for the teachers to use digital image data to develop classification models. My design framework, with an embedded approach of drawing on familiar statistics ideas first before extending these ideas into less familiar territory, appeared to support the teachers' development of new statistical and computational ideas related to algorithmic modelling.

In this chapter, I explored the introduction of code-driven tools through a classification modelling task (Task 1) and demonstrated how the second iteration of my task design framework was aligned to this task. In Chapter 7, I explore the introduction of code-driven tools through a simulation-based probability modelling task (Task 4) and use the task and its implementation with teachers to evaluate and refine the task design framework and examine integrated statistical and computational thinking.

Chapter 7: Probability modelling task

7.1 Introduction

Recommendations for the implementation of data science at the high school level include providing opportunities for students to integrate both computational and statistical thinking (e.g., De Veaux et al., 2017). The teaching of probability simulations offers opportunities for students to use a variety of digital tools to co-develop statistical and computational knowledge. For instance, there are *code-driven* tools, computational tools which users interact with predominantly by entering and executing text commands (code) and there are *GUI-driven* tools, computational tools which users interact with predominantly by pointing, clicking, or gesturing. Although technology-enabled simulations have been used to introduce students to probability and statistical inference since the earliest days of computers in classrooms (e.g., Simon et al., 1976; Thomas & Moore, 1980), advances in digital technologies have enabled the development of intuitive tools for supporting and visualising results (e.g., Ben-Zvi, 2000).

Modern tools such as spreadsheets, purpose-built statistical GUI-driven software, and programming languages are all possibilities for teaching probability simulations. Barriers for broader implementation of digital tools for teaching probability simulation, however, include teacher confidence with using these tools to articulate probability models, generate data from these models, summarise and analyse the simulated data, and visualise the simulated data. Because there is limited research in data science education for supporting teachers to move to, and between, different computational tools for teaching statistical modelling, my research seeks to explore how different tools can support or stimulate thinking about probability simulations and new computational ideas.

Using a DBR approach, I explicated an initial task design framework to introduce learners to code-driven-tools through statistical modelling in Chapter 4, which was refined in Chapter 5 and corroborated in Chapter 6. In this chapter, I explore the design and implementation of a task for introducing teachers to code-driven tools for simulation-based probability modelling (Task 4). I ascertain if the task design framework can be applied for constructing a task that uses a different statistical modelling situation (probability simulation) and unfamiliar computational tools (both GUI-driven and code-driven). I also explore what statistical and computational thinking practices emerge when the task is implemented with teachers.

7.2 Teaching probability simulations at the high school level

Traditional approaches to teaching probability have focused on calculating single probabilities or expected values of random variables (Maxara & Biehler, 2006), often using coins, spinners, and dice to demonstrate theoretical probability (Pratt, 2011). Consequently, teachers can associate simulation activities with a frequency view of probability and not consider the use of simulation to explore distribution as an important concept (e.g., Sánchez, 2002). Probability simulations can be used to develop conceptual understanding of probability, in particular the notions of randomness and its effect on variation and distribution (Engel, 2010; Pfannkuch et al., 2016; Watkins et al., 2014). With respect to randomness, students can be encouraged to interact with chance devices and visualise outcomes being generated dynamically (Pfannkuch & Budgett, 2016), and with learning focused on the random process not just the product of the simulation (Kaplan et al., 2014b).

Simulations can be used to solve difficult probability problems, providing a simpler approach than using difficult analytical methods (Biehler, 1991). Research has implored educators to focus on the use of simulation to promote a modelling perspective to teaching statistics (Garfield et al., 2012), as central to statistical thinking is the use of statistical models (Wild & Pfannkuch, 1999). As a probability simulation involves replacing a real random situation with a model, which can be manipulated and used to generate data that can be analysed (Engel, 2010), there is potential to adopt a greater focus on the model when teaching probability simulations. Teachers could encourage students to see and apply structure as part of the probability modelling process (e.g., Pfannkuch et al., 2016), and to explore "what if?" scenarios by changing features of the probability model being used for the simulation. When using technology for probability modelling, learners can quickly run multiple simulations and explore the relationship between the model and the data it generates (Fergusson & Pfannkuch, 2020; Kazak & Pratt, 2021). Examples of existing tasks that demonstrate the exploration of a model within a simulated environment include a basketball simulation (Prodromou, 2014) and data games (Erickson, 2013). A focus on modelling for teaching probability simulations is also consistent with recommendations for teaching data science at the high school or introductory level (e.g., Fries et al., 2021).

To support students to use technology to conduct probability simulations, they need to be provided with hands-on tactile experiences with the processes that will be used by the technology, so that their understanding can be connected to what the computer is doing (e.g., Chance et al., 2004; Erickson, 2006; Gould et al., 2010; Pfannkuch et al., 2013). Using "unplugged" simulations supports students to use physical actions to represent steps in the modelling process (Wood, 2005), which can benefit student understanding when they move to using computational tools (e.g., Chance & Rossman, 2006; Zhang et al., 2021). Technology which provides "virtual" versions of physical chance devices such as coins and spinners has also been described as critical, so that students view the computer models they build as generating the same data as they would obtain using the actual devices (Konold & Kazak, 2008). Ideally, the computational tools used to carry out simulations would place the focus on defining the model, not the physical or time-consuming tasks of generating each trial manually. In this way, technology is used as a mechanism to shift statistics from being focused on computations, formulae and procedures towards a learning culture that supports informal and open exploration of data and models with minimal barriers (e.g., Ben-Zvi, 2000; Wild, 2018).

Statistics education researchers have carefully considered how GUI-driven tools influence students' thinking when learning statistics and have also considered how to define the complex relationship between software features, task design, and learners' statistical conceptions (e.g., Ben-Zvi, 2000; Biehler, 1997b). For instance, guidelines developed by delMas (1997) for the use of technology in the teaching of statistics present many pedagogical perspectives on the use of simulations, such as the need to provide clear examples of how models are built and interpreted, and the need for physical activities as well as computer simulations. Considering the relationship between tool, task, and thinking for modelling activities is also important (Biehler, 2018; Doerr & Pratt, 2008; Moore, 1997). The design of the task should consider the design of the computational tool and allow students to tinker with a model and to visualize changes instantaneously. For instance, *Tinkerplots* (Konold & Miller, 2015) has been used extensively by statistics education researchers to explore probability simulations and simulation-based inference (e.g., Noll & Kirin, 2016; Wright et al., 2019). Furthermore, Wild et al. (2017) advised teachers to focus on what students are seeing and thinking when interacting with software, and to ask students to explain and describe the key elements of what they are doing with the tool.

GUI-driven tools dominate the teaching of statistics at the high school level, yet it is difficult to find substantial literature that explicitly communicates strategies for designing tasks that introduce code-driven tools for probability simulations, or more broadly statistical modelling. In a research study conducted by Ferreira et al. (2014), high school students successfully used the programming language *R* (R Core Team, 2020) to conduct simulations after completing "by-hand" probability activities. They reported that the students quickly became familiar with the language and experienced a sense of control in being able to manipulate the computational actions. Code-driven tools could lower the cognitive demands of statistical modelling tasks (e.g., Son et al., 2021) and assist the teaching of probability simulations, although it is not clear how teachers would balance learning new statistical and computational knowledge within the same task. For example, when encountering issues with executing code, learners might frame problems as either statistical or computational (Thoma et al., 2018).

Advice for using technology to teach statistics has included that a combination of tools may be best for student learning (e.g., Chance et al., 2007; Hesterberg, 1998). When selecting and combining different computational tools for teaching probability simulations, however, considerations are needed about how to support students' mental images and visual representations for each aspect of the modelling process. According to dual coding theory, humans store information in their memory in two distinct representations, verbal and non-verbal, and employing both representations enhances learning (e.g., Clark & Paivio, 1991). Using both GUI-driven and code-driven tools allows learners to interact with text-based and graphics-based representations of statistical and computational concepts (e.g., Cook & Goldin-Meadow, 2006). According to Erickson et al. (2019), the naming of actions that alter a data set's contents or structures as "data moves" might help students move between GUI-driven and code-driven tools. Similarly, I propose that the closer the match between modelling actions carried out with GUI-driven and code-driven tools, the smoother the move for the learner. However, there is limited research on high school teachers or students using both GUI-driven and code-driven tools within the same statistics learning programme (e.g., Biehler & Fleischer, 2021; Deitrick et al., 2017; Thoma et al., 2018).

7.2.1 The New Zealand teaching context

Probability simulations are included in the assessed curriculum for statistics students in their last two years of high school. A common situation used for teaching and assessing probability simulations is the "cereal box problem." The "cereal box problem" involves a certain number of different "cards" that can be collected from "cereal boxes", with the goal to collect a full set of "cards" (one of each different card). Students are typically asked to carry out a probability simulation to estimate the mean number of "cereal boxes" that need to be collected to obtain a full set of "cards." Two of the three national exemplar assessment tasks provided to teachers in New Zealand are based on the "cereal box problem": *Fruity freezes* and *Oh what a seed!* For the *Fruity freezes* task ice blocks (popsicles) are used instead of cereal boxes. There are four symbols that are printed on the ice block sticks (apple, pineapple, grape, strawberry) and each symbol has a different probability (0.4, 0.3, 0.2, 0.1). The *Fruity freezes* task includes an additional stopping clause, which is that a person called "Grace" will only buy one ice block per day for up to 10 days.

The *Fruity freezes* national exemplar assessment does not demonstrate the use of GUI-driven or code-driven tools for teaching probability simulations. Instead, it demonstrates the use of a calculator to generate random numbers, each of which are mapped to a particular symbol, and a "by hand" approach to recording and summarising the data to calculate the mean number of days ice blocks are bought. The data generated during the simulation are recorded in one table, where each row represents a different trial. The exemplar does not show the data generated visually as a distribution, which would allow for greater discussion of the variation of the outcomes under chance. Because teachers are familiar with the exemplar, I decided to design a task that explored the same *Fruity freezes* situation with new GUI-driven and code-driven tools.

7.3 Task design framework

Through retrospective analysis of three other tasks (Chapters 4, 5, & 6), I explicated a design framework to introduce learners to code-driven tools through statistical modelling. Because design-based research is an iterative process, I retrospectively analysed the second iteration of the task design framework (Chapter 5) against the criterion of producing a more general design framework rather than one that was too specific to the nature of the tools and tasks used previously. For the *third iteration* of the task design framework, changes were made to two design considerations: the data used (C2) and the level of computational transparency (C4).

In previous iterations of the design framework, the *data used* (C2) design consideration referred to "sets of data." However, the use of data in the other tasks included data sourced dynamically, for example from APIs (Application Programming Interfaces). Therefore, the *data used* (C2) design consideration now refers to "sources of data." The *level of computational transparency* (C4) design consideration in previous iterations of the design framework did not explicate the need to select or develop features of computational tools. This directive has now been added, as the task designer may often need to modify the interface of the GUI-driven tool or develop new functions for a code-driven tool.

Table 7.1 provides the modified design considerations for the *third iteration* of the task design framework.

Table 7.1: The four design considerations of the design framework

Design consideration	Related design decisions
Introduction of new knowledge (C1)	Identify content and use a sequence of phases and steps within a task to introduce learners to new ideas
Data used (C2)	Select and use different variables or different sources of data within the same data context for the task
Tools used (C3)	Combine different tools for statistical modelling (unplugged, code-driven, GUI-driven) and connect actions or representations between tools within the task
Level of computational transparency (C4)	Select or develop features of computational tools with respect to how obvious the computations performed by the tool are to the learner

7.4 Task design characteristics

I now discuss how the *third iteration* of my task design framework explicates the key characteristics of the task used for the research reported in this paper. The statistical modelling approach for the task involved: adapting probability models represented using technology; using these models to generate data through simulation; and using the model-generated data to calculate estimates of model-related values, for example, probabilities or expected values. The learning goal for the task was for teachers to use a code-driven tool and probability simulations to develop a probability model that would inform the requirements for a fictional "card collecting" competition. The *data used* (C2) were generated through simulation from probability models based on two of the national exemplar assessment tasks provided to teachers in New Zealand, *Fruity freezes* and *Oh what a seed!*

7.4.1 Tool-related design decisions

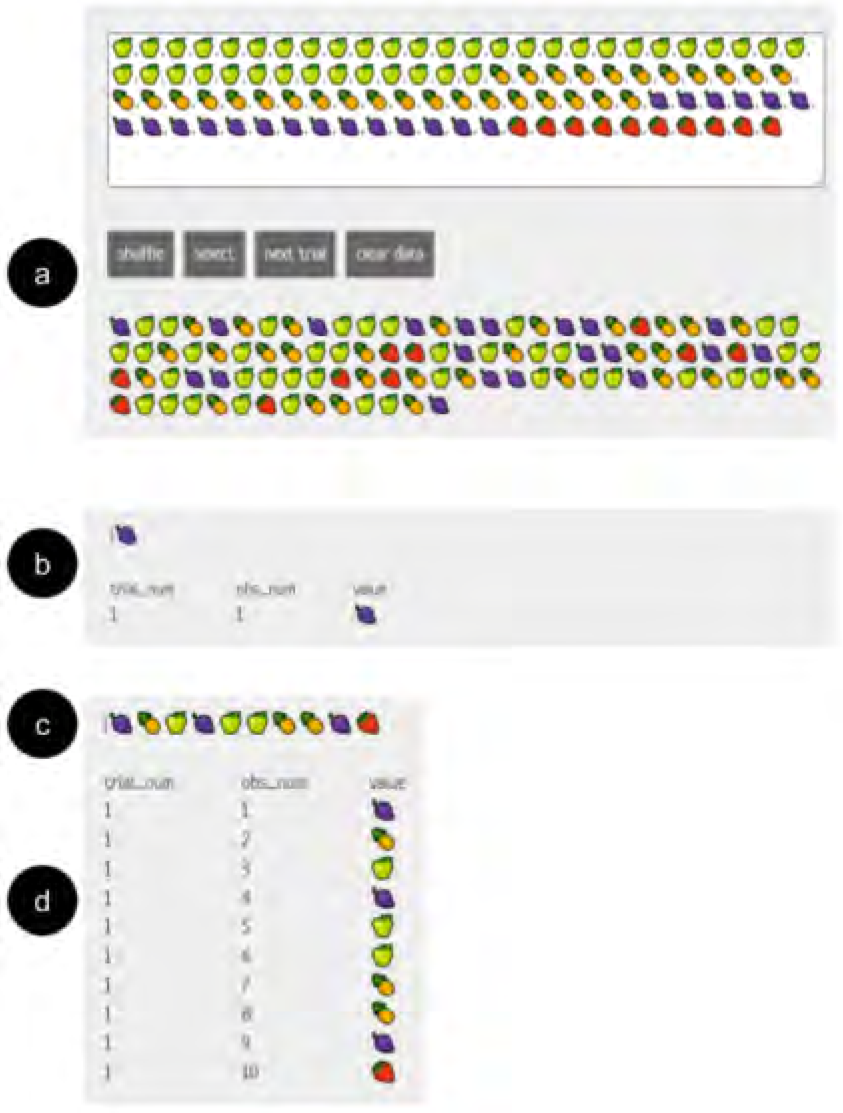
Unlike the previous three tasks, I could not assume that the teachers had experience with using a GUI-driven tool for conducting probability simulations and visualising results graphically.

Additionally, the *unplugged* approach demonstrated in the national exemplar assessment represented the data generated in one table, where each row was one trial, rather than a hierarchal data structure. Therefore, there were two tool-related task design challenges that needed to be resolved: (1) the selection and sequencing of computational tools to move students towards a code-driven tool, and (2) the introduction of an unfamiliar structure, hierarchal, for representing the data generated. In line with the design consideration of *tools used* (C3), which recommends connecting actions or representations between different tools, the task introduced three different computational tools to explore the same probability simulation problem: (1) the *Simple Sampler* tool, a new online application that I developed specifically for this research; (2) the software tool *CODAP* (codap.concord.org) using a modification of the existing *Sampler* plugin; (3) the programming language *R*, via a web page created using the *R* package *learnr* (Schloerke et al., 2018) that allowed code to be run in the browser. As the computational tools were unfamiliar to teachers, the design consideration of *introduction of new knowledge* (C1) informed the development and modification of features of the tools simultaneously with the design consideration of *level of computational transparency* (C4).

At the core of the design principles for the framework is that learners are *familiar* with a statistical modelling process (P2) and can *describe* the computational steps of the process based on experiences with non-code-driven tools (P3). *CODAP* was selected as the GUI-driven tool to support the teachers to move from an *unplugged* approach to a code-driven approach, as the tool provided actions or representations for the modelling steps needed to design, carry out and interpret the results from a simulation. Additionally, *CODAP* provided (1) a sampler plugin that automated and visualised the random generation of outcomes (C3), and (2) interactive tables of data that could be manipulated using formulae in similar ways to the functions the teachers would meet in the programming language *R* (C4). The *CODAP Sampler* plugin, however, did not allow for a trial to end when one of each outcome (or symbol) had been generated, one of the conditions for the *Fruity freezes* task. Consequently, I developed new code to modify and produce an adapted *Sampler* plugin that could be used for the teaching experiment (C4). A feature of the data generated from the *CODAP Sampler* plugin was its organization into multiple tables in a hierarchal structure, a data structure that was unfamiliar to the teachers. As I began to develop materials for the task using *CODAP*, I realised there would be too much new information to grasp at the same time if teachers moved from an *unplugged* approach directly to *CODAP* (C1).

Therefore, the decision was made to create a new GUI-driven tool called the *Simple Sampler*.

The design of the *Simple Sampler* was based on the key computational representations, actions and words needed to support and describe the computational steps for a probability simulation using *CODAP* and the programming language *R* (C3). Figure 7.1 shows the graphical user interface of the *Simple Sampler* tool and the use of the tool to carry out one trial.



trial_num	obs_num	value
1	1	🟪

trial_num	obs_num	value
1	1	🟪
1	2	🟧
1	3	🟩
1	4	🟪
1	5	🟩
1	6	🟩
1	7	🟧
1	8	🟧
1	9	🟪
1	10	🟥

Figure 7.1: An example of using the Simple Sampler tool to carry out one trial with annotations showing key features of the graphical user interface and visualisations

The *Simple Sampler* tool was designed to visualise the probability model for the simulation in a very similar way to the *Sampler* plugin for *CODAP* (C4) and provided four different buttons for teachers to use: shuffle, select, next trial, clear. The "shuffle" button replaced all sampled items and shuffled them (Figure 7.1a). The "select" button selected the first shuffled item (Figure 7.1b), and if the "select" button was then pressed again, it took the second shuffled item, and so on. Visually, the value from each selection was shown horizontally as a row (Figure 7.1c), to match to how teachers were familiar with recording the values of one trial from a simulation using an *unplugged* approach (C3). Simultaneously, each value was shown in a table at the bottom of the page (Figure 7.1d), where each row represented a new value (obs_num), and an additional column that kept track of the trial number (trial_num). When the "next trial" button was pressed, a new row was added to the horizontal presentation of the values (not shown), and the trial number increased by one to the table representation of the values. In this way, the tool supported the teachers to make connections between the different data structures and those produced by *CODAP* (C3).

The use of *CODAP* for the task also required careful consideration, particularly with respect to computational transparency (C4). At the time of the teaching experiment, there was no function/formula provided by the tool that could be used to determine how many unique values there were in the results from one trial, an essential requirement for the *Fruity freezes* problem. Therefore, the decision was made to demonstrate the pre-prepared "finished product" of using *CODAP* for the *Fruity freezes* simulation, rather than engage the teachers with the complexities of the formulae used within the summary data table. In this way, the structure of the summary data table could be the focus and could be used to support the teachers to make connections between the names of variables and the sequence of calculations and logic needed to determine when a trial ends and whether a prize has been won or not (C3). The *CODAP* tool was also used to introduce a graphical approach for calculating the model-generated estimates required for the *Fruity freezes* problem (C1). For example, the probability of winning a prize was calculated by constructing a bar chart and reading the proportion for the outcome "yes."

Both the *Simple Sampler* and *CODAP* tools provided teachers with the computational actions and representations for the key modelling steps needed to make connections to the *R* code used in the code-driven tool, the *learnr* created web page (C3). The *tidyverse* (Wickham, 2017) ecosystem of *R* packages was used for the code, with additional functions that I developed to assist the readability of the code and plotting functions from the *R* package *iNZightPlots* (Elliott et al., 2021). For example, the visualise function was written to enable teachers to visualise the probability model for the simulation in a very similar way to the *CODAP Sampler* plugin and the *Simple Sampler* tool (C4). The names assigned to objects used within the *R* code were also carefully matched to the labels and words used in both the *Simple Sampler* and *CODAP* tools. Additionally, emojis were used within the code-driven tool, as they had been across all tools, to represent the outcomes for the probability model.

7.4.2 Summary of task phases

Each of the six phases of the task is now described (see also Appendix E). For each phase, the description begins with the relevant design principle, related learning action or experience, and anticipated learning from the *third iteration* of my task design framework.

Phase 1

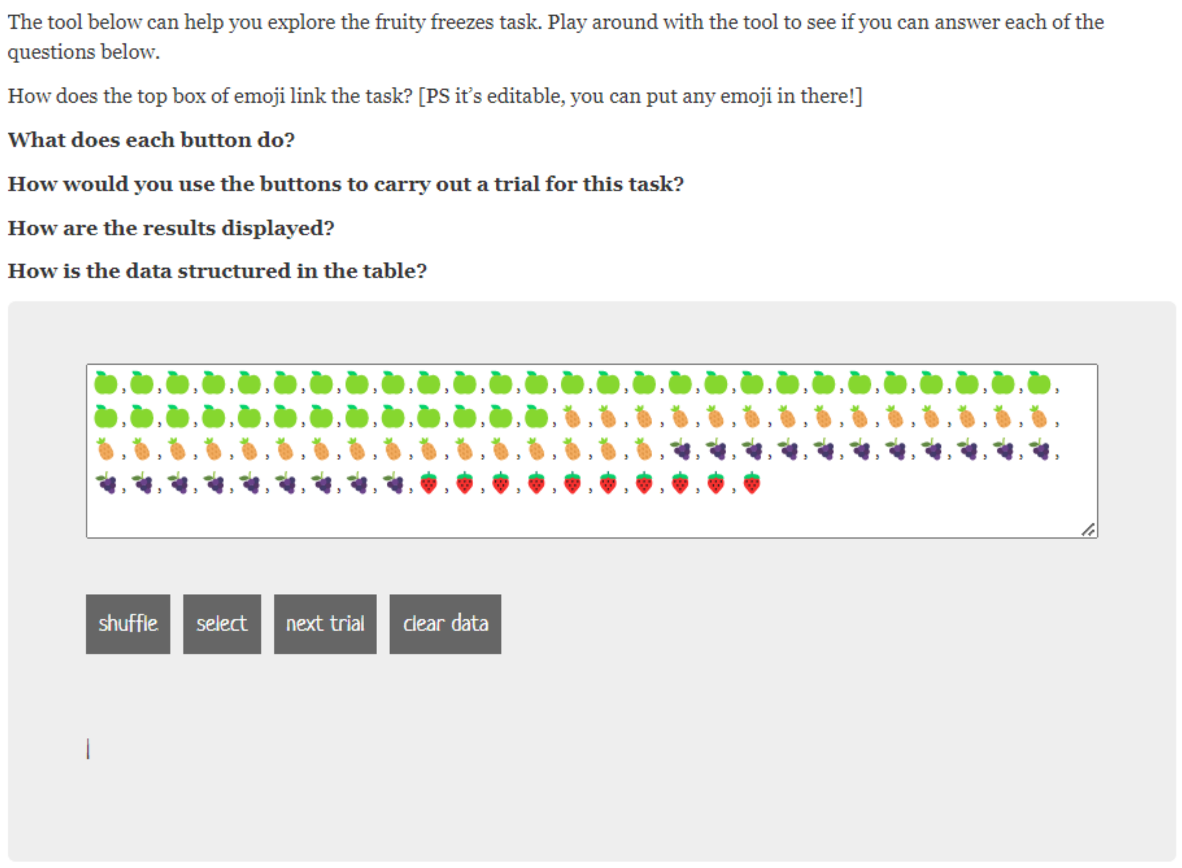
Immerse in data context (P1) | Participate in activities that promote engagement with the data context | Understanding the nature of the data that is used across the task

The teachers were introduced to the modelling context of people collecting objects. They were shown a video explaining some reasons why people collect objects and then they were provided with examples of recent media articles discussing current "collecting promotions" run by a supermarket and a cereal food company. The teachers were asked to discuss the following questions: *What drives people to become collectors? What are your experiences with collecting? Have you ever collected cards/tiles/etc. as part of competition or promotion?*

Phase 2

Re-familiarise with statistical modelling ideas (P2) | Carry out familiar statistical modelling activities without using code | Application of statistical thinking

The teachers were given a paper copy of the *Fruity freezes* task and asked to discuss with each other how they would teach students to complete the task. The teachers then moved to a web page and were given access to the *Simple Sampler* tool to carry out the simulation. Figure 7.2 shows the instructions and graphical user interface of the *Simple Sampler* tool.


The tool below can help you explore the fruity freezes task. Play around with the tool to see if you can answer each of the questions below.

How does the top box of emoji link the task? [PS it's editable, you can put any emoji in there!]

What does each button do?

How would you use the buttons to carry out a trial for this task?

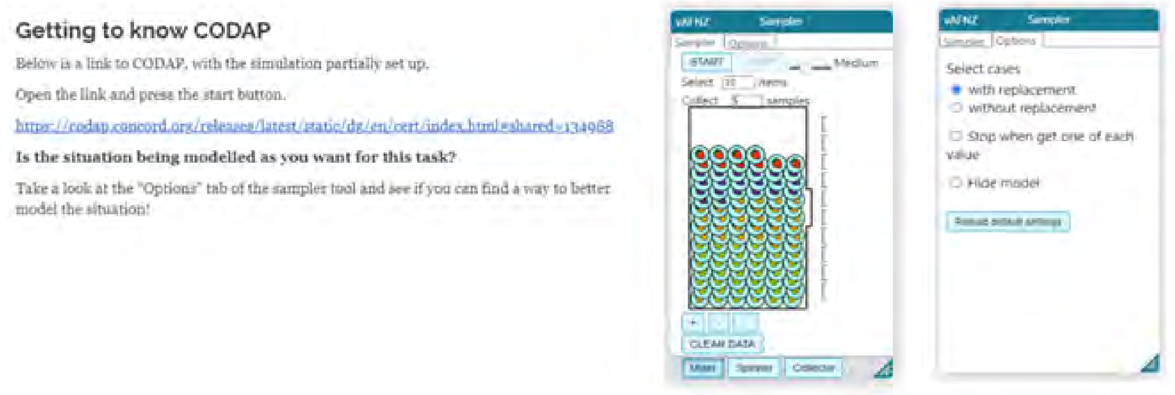
How are the results displayed?

How is the data structured in the table?

🍏,
🍏,🍏,🍏,🍏,🍏,🍏,🍏,🍏,🍏,🍏,🍏,🍏,🍏,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,
🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍍,🍇,🍇,🍇,🍇,🍇,🍇,🍇,🍇,🍇,🍇,🍇,
🍇,🍇,🍇,🍇,🍇,🍇,🍇,🍇,🍇,🍓,🍓,🍓,🍓,🍓,🍓,🍓,🍓,🍓,🍓

Figure 7.2: The task instructions and graphical user interface for the Simple Sampler tools

The tool *CODAP* was then introduced to the teachers using two links to pre-prepared files with the simulation part-way through, Figure 7.3 and Figure 7.4 respectively.


Getting to know CODAP

Below is a link to CODAP, with the simulation partially set up.

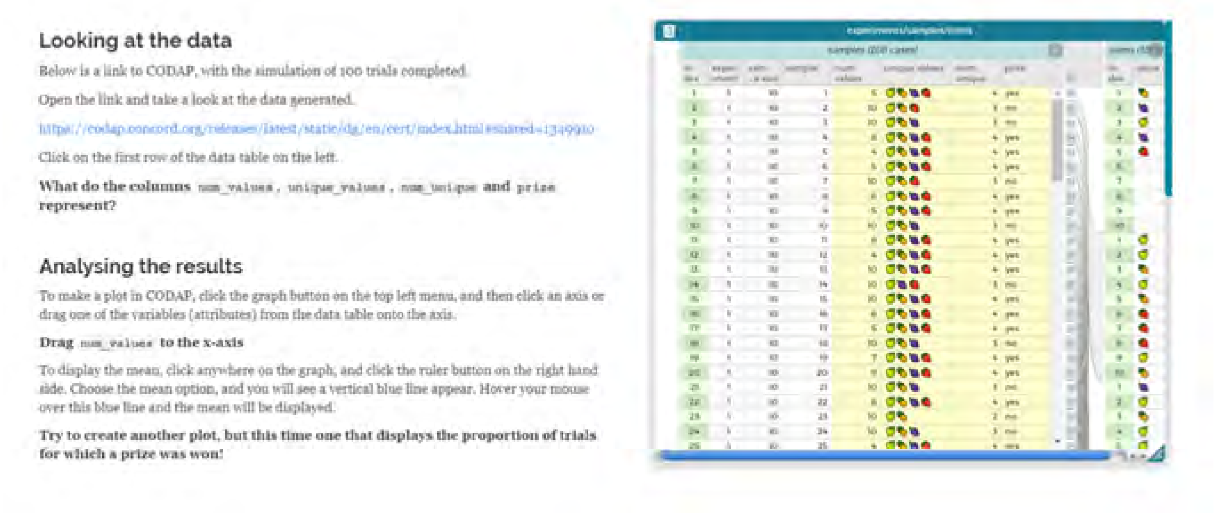
Open the link and press the start button.

https://codap.concord.org/releases/latest/static/dg/en/cert/index.html#shared=134988

Is the situation being modelled as you want for this task?

Take a look at the "Options" tab of the sampler tool and see if you can find a way to better model the situation!

Figure 7.3: The first pre-prepared CODAP file showing the modified Sampler plugin and the task instructions for considering how the situation is being modelled



Looking at the data

Below is a link to CODAP, with the simulation of 100 trials completed.

Open the link and take a look at the data generated.

https://codap.concord.org/releases/latest/static/dg/en/cert/index.html#shared=134991

Click on the first row of the data table on the left.

What do the columns `num_values`, `unique_values`, `num_unique` and `prize` represent?

Analysing the results

To make a plot in CODAP, click the graph button on the top left menu, and then click an axis or drag one of the variables (attributes) from the data table onto the axis.

Drag `num_values` to the x-axis

To display the mean, click anywhere on the graph, and click the ruler button on the right hand side. Choose the mean option, and you will see a vertical blue line appear. Hover your mouse over this blue line and the mean will be displayed.

Try to create another plot, but this time one that displays the proportion of trials for which a prize was won!

experiments/samples/items

samples (200 cases)

index	experiment	sample_size	sample	num_values	unique_values	num_unique	prize
1	1	10	1	5	🍏🍊🍇🍓	4	yes
2	1	10	2	10	🍏🍊🍓	3	no
3	1	10	3	10	🍏🍊🍇	3	no
4	1	10	4	8	🍏🍊🍇🍓	4	yes
5	1	10	5	4	🍏🍊🍇🍓	4	yes
6	1	10	6	5	🍏🍊🍇🍓	4	yes
7	1	10	7	10	🍏🍊🍓	3	no
8	1	10	8	8	🍏🍊🍇🍓	4	yes
9	1	10	9	5	🍏🍊🍇🍓	4	yes
10	1	10	10	10	🍏🍊🍇	3	no
11	1	10	11	8	🍏🍊🍇🍓	4	yes
12	1	10	12	4	🍏🍊🍇🍓	4	yes
13	1	10	13	10	🍏🍊🍇🍓	4	yes
14	1	10	14	10	🍏🍇🍓	3	no
15	1	10	15	10	🍏🍊🍇🍓	4	yes
16	1	10	16	8	🍏🍊🍇🍓	4	yes
17	1	10	17	5	🍏🍊🍇🍓	4	yes
18	1	10	18	10	🍏🍊🍇	3	no
19	1	10	19	7	🍏🍊🍇🍓	4	yes
20	1	10	20	9	🍏🍊🍇🍓	4	yes
21	1	10	21	10	🍏🍊🍇	3	no
22	1	10	22	8	🍏🍊🍇🍓	4	yes
23	1	10	23	10	🍏🍊	2	no
24	1	10	24	10	🍏🍊🍇	3	no
25	1	10	25	4	🍏🍊🍇🍓	4	yes

items

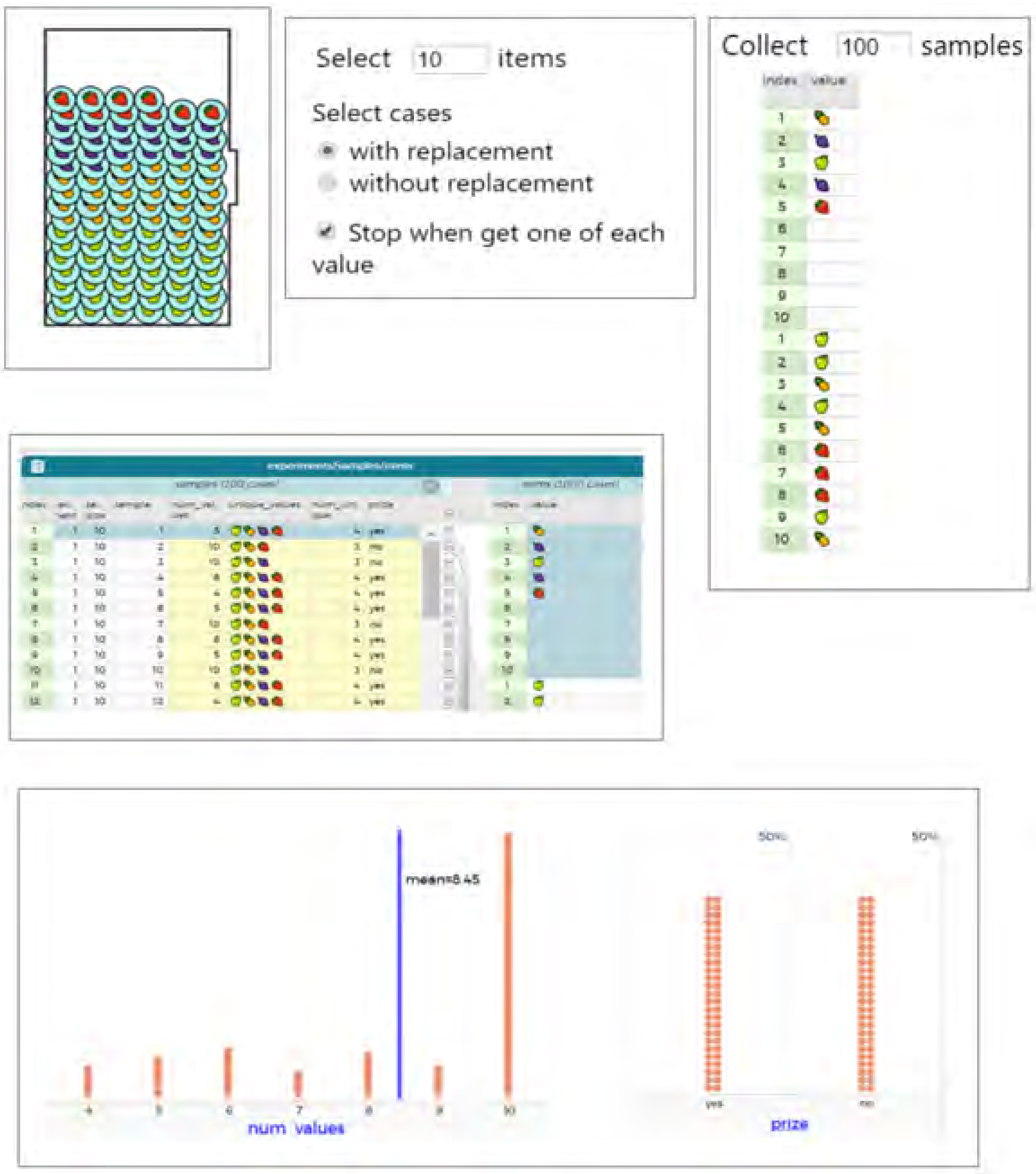
index	value
1	🍊
2	🍇
3	🍏
4	🍇
5	🍓
6	
7	
8	
9	
10	
1	🍏
2	🍏
3	🍊
4	🍏
5	🍊
6	🍓
7	🍓
8	🍓
9	🍏
10	🍊
1	🍇
2	🍏
3	🍊
4	🍏
5	🍏

Figure 7.4: The second pre-prepared CODAP file and the task instructions about what to look at in the data and how to analyse the results

Phase 3

Describe computational steps of statistical modelling process (P3) | Use words to describe key computational steps of statistical modelling process | Decomposition of modelling steps and recognising required computation

The teachers were given five individual pieces of paper, each showing a modelling step in the form of screenshots taken from *CODAP* to model the *Fruity freezes* situation (Figure 7.5). They were asked to arrange the screenshots in the order that they would use *CODAP* to investigate the *Fruity freezes* task. In the space beside each screen shot/step, the teachers were asked to write brief notes about what was happening statistically and/or computationally, and how this linked to the *Fruity freezes* task.



Select 10 items

Select cases
- [✓ with replacement]
- without replacement

[✓ Stop when get one of each value]

Collect 100 samples

index	value
1	🟠
2	🟣
3	🟢
4	🟣
5	🔴
6	
7	
8	
9	
10	
1	🟢
2	🟢
3	🟠
4	🟢
5	🟠
6	🔴
7	🔴
8	🔴
9	🟢
10	🟠

experiments/samples/items

samples (100 cases)

index	exp-eriment	sa-size	sample	num_values	unique_values	num_unique	prize
1	1	10	1	5	🟢🟠🟣🔴	4	yes
2	1	10	2	10	🟢🟠🔴	3	no
3	1	10	3	10	🟢🟠🟣	3	no
4	1	10	4	8	🟢🟠🟣🔴	4	yes
5	1	10	5	4	🟢🟠🟣🔴	4	yes
6	1	10	6	5	🟢🟠🟣🔴	4	yes
7	1	10	7	10	🟢🟠🔴	3	no
8	1	10	8	8	🟢🟠🟣🔴	4	yes
9	1	10	9	5	🟢🟠🟣🔴	4	yes
10	1	10	10	10	🟢🟠🟣	3	no
11	1	10	11	8	🟢🟠🟣🔴	4	yes
12	1	10	12	4	🟢🟠🟣🔴	4	yes

items (1000 cases)

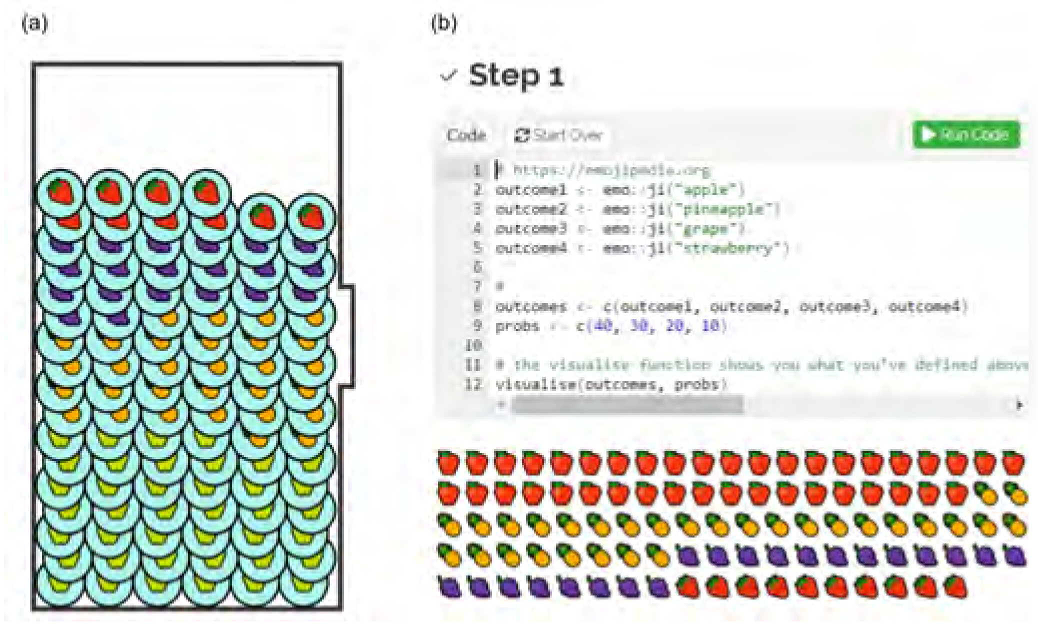
index	value
1	🟠
2	🟣
3	🟢
4	🟣
5	🔴
6	
7	
8	
9	
10	
1	🟢
2	🟢

Figure 7.5: The five screenshots taken from CODAP to model the Fruity freezes situation

Phase 4

Match statistical modelling steps to code chunks (P4) | Read and match lines of code with statistical modelling steps | Recognising aspects of code syntax and structure

The teachers were progressively given chunks of code that matched each of the five modelling steps presented in Phase 3. The teachers were asked to match each code chunk to one of the screenshots from Phase 3 and to add short comments to the code to describe what they thought the code did. Figure 7.6a shows the screenshot for the first modelling step and Figure 7.6b the first code chunk revealed to the teachers.



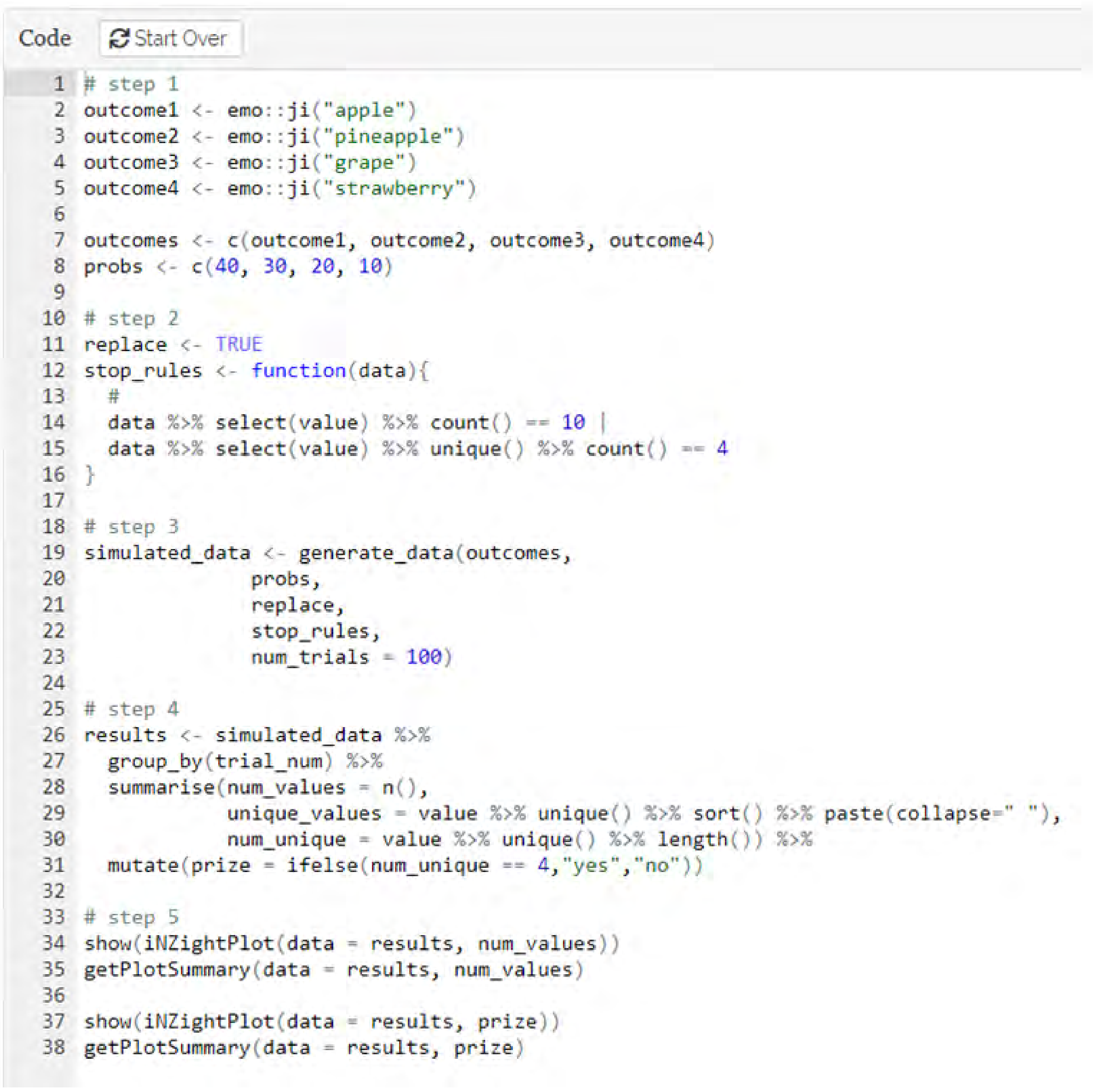
```
# https://emojipedia.org
outcome1 <- emo::ji("apple")
outcome2 <- emo::ji("pineapple")
outcome3 <- emo::ji("grape")
outcome4 <- emo::ji("strawberry")

#
outcomes <- c(outcome1, outcome2, outcome3, outcome4)
probs <- c(40, 30, 20, 10)

# the visualise function shows you what you've defined above
visualise(outcomes, probs)
```

Figure 7.6: (a) The screenshot of the first modelling step, (b) The first code chunk revealed to the teachers

Phase 5

Adapt code chunks with slight modifications (P5) | Identify features of code to change to complete a statistical modelling action | Integration of statistical and computational knowledge

The teachers were given a paper copy of the *Oh what a seed!* task, a very similar situation to the *Fruity freezes* task, which involved a gardening store running a promotion for customers to collect different seed packets (C2). The teachers were asked to adapt the code provided to model the new situation. All the code used from Phase 4 was presented to the teachers (Figure 7.7).



```r
# step 1
outcome1 <- emo::ji("apple")
outcome2 <- emo::ji("pineapple")
outcome3 <- emo::ji("grape")
outcome4 <- emo::ji("strawberry")

outcomes <- c(outcome1, outcome2, outcome3, outcome4)
probs <- c(40, 30, 20, 10)

# step 2
replace <- TRUE
stop_rules <- function(data){
  #
  data %>% select(value) %>% count() == 10 |
  data %>% select(value) %>% unique() %>% count() == 4
}

# step 3
simulated_data <- generate_data(outcomes,
                              probs,
                              replace,
                              stop_rules,
                              num_trials = 100)

# step 4
results <- simulated_data %>%
  group_by(trial_num) %>%
  summarise(num_values = n(),
            unique_values = value %>% unique() %>% sort() %>% paste(collapse=" "),
            num_unique = value %>% unique() %>% length()) %>%
  mutate(prize = ifelse(num_unique == 4,"yes","no"))

# step 5
show(iNZightPlot(data = results, num_values))
getPlotSummary(data = results, num_values)

show(iNZightPlot(data = results, prize))
getPlotSummary(data = results, prize)
```

Figure 7.7: The R code provided to teachers in Phases 4 (progressively), 5 and 6

Phase 6

Explore ”what if?” changes to code (P6) | Modify at least one aspect of provided code to produce new or unexpected outputs | New knowledge gained by integrating statistical and computational thinking

The teachers were again provided with all the code used in Phase 4 for the *Fruity freezes* task (Figure 7.7). They were asked to use the code-driven tool to explore three “what if?” questions: *What if Grace only buys ice blocks for five days?*, *What if there was no limit on how many days Grace buys ice blocks for?* and *How does Grace’s chances of winning a prize change as the number of days she buys ice blocks for increases?* Finally, they were asked to complete the challenge where they had to create a new version of the competition with two prizes: one that over 90% of customers could win if they collected 20 cards and one that no more than 20% of customers could win if they collected 20 cards. The teachers had to decide how many "cards" would be used, the probabilities for each card and how two different prizes would be determined.

7.5 Analysis

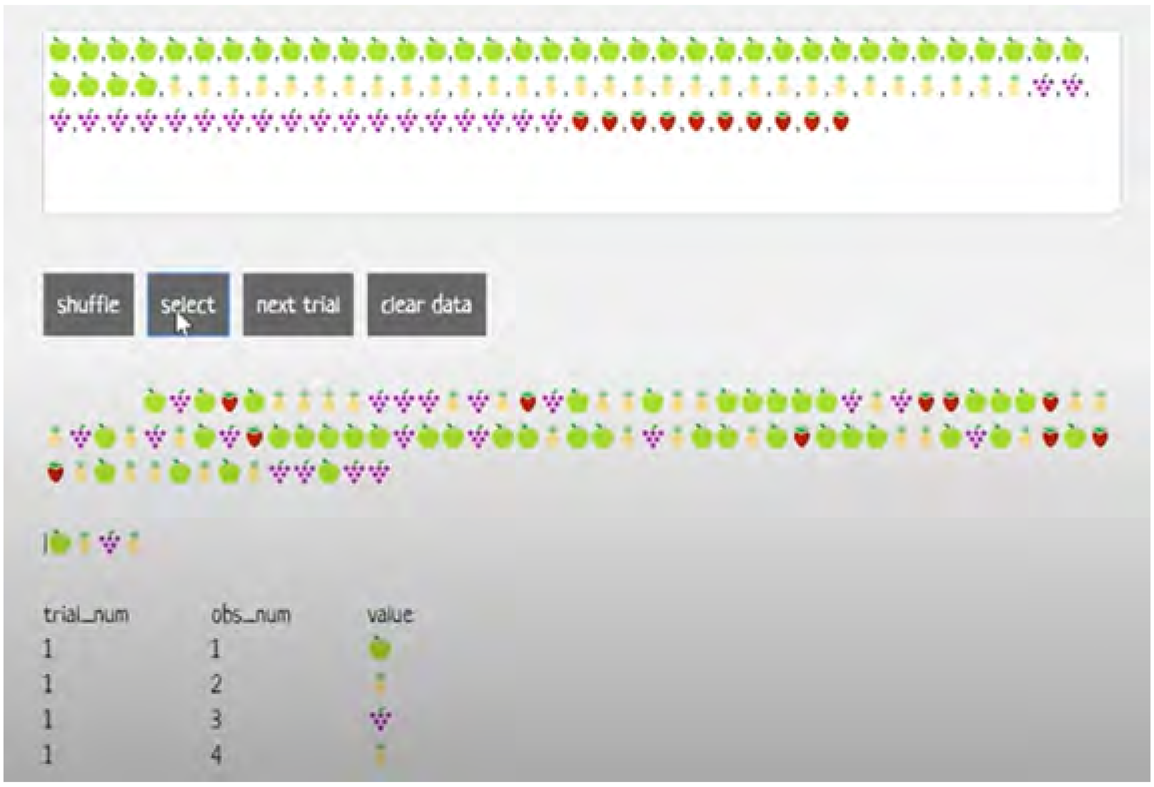
The implementation of the probability modelling task took place during the fourth day of the professional development workshops. Four of the ten teachers had participated in the earlier workshops and six had not. The teacher pairings for the task were: Amelia and Harry; Erika and Laina; Naomi and Umika; Ella and Yin; and Alice and Matiu (pseudonyms have been used). Due to technical issues, the last pair of teachers was not recorded.

Four themes were identified with respect to teachers' statistical and computational thinking practices related to probability simulations: *reconciling prior tool thinking*, *prior task-frame thinking*, *tool-stimulated thinking*, and *task-stimulated thinking.* For each theme, I focus on one pair of teachers to illustrate common actions and reasoning.

7.5.1 Reconciling prior tool thinking

I observed that when the task introduced the new computational tools, the teachers did not seem to activate or transfer their existing statistical thinking practices related to probability simulations. Because their thinking appeared to be driven by familiar computational tools, I described the theme as *reconciling prior tool thinking.* An example of reconciling prior tool thinking was related to how random outcomes can be generated from a probability model.

In Phase 2 of the task, the teachers discussed using the random number function available on commonly used calculators or spreadsheets to carry out a probability simulation for the *Fruity freezes* task. For instance, Amelia and Harry discussed how to use a random number function to generate a value between 1 and 10 and how to "map" this value to one of the outcomes of the probability model. They discussed that if a 1, 2, 3, or 4 was generated, then it would be recorded as an apple, as the symbol apple had a probability of 0.4. However, in Phase 3 when the *Simple Sampler* tool was introduced, Amelia and Harry initially sampled without replacement when generating outcomes for each trial. Figure 7.8 provides a screenshot of how they used the tool to randomly generate four symbols without replacement using the *Simple Sampler* tool.



trial_num	obs_num	value
1	1	🍏
1	2	🍍
1	3	🍇
1	4	🍍

Figure 7.8: A screenshot of Amelia and Harry using the Simple Sampler tool to randomly generate four symbols without replacement

Sampling without replacement from a set of 100 emoji meant that the probabilities of each symbol that the emojis represented did not remain constant. The tool the teachers were familiar with, the random number function on a calculator or spreadsheet, does not require a decision to be made about the method of sampling, as it is always samples with replacement. Additionally, the random number function on a calculator or spreadsheet provides no visual imagery of *how* the number is randomly selected or generated. In fact, Amelia discussed with Harry her concern with using spreadsheets for teaching simulation ideas for similar reasons stating, “the connection is often lost when you are simulating the process by using the spreadsheet, it’s random you know.”

In contrast, the *Simple Sampler* tool appeared to highlight randomness to the teachers, as Amelia and Harry connected the shuffle button to the visual shuffling of the emojis, concluding this shuffling process produced random outcomes. Also, it appeared the teachers were unfamiliar with using objects as probability models and the additional thinking and computational actions using objects required, as Amelia described:

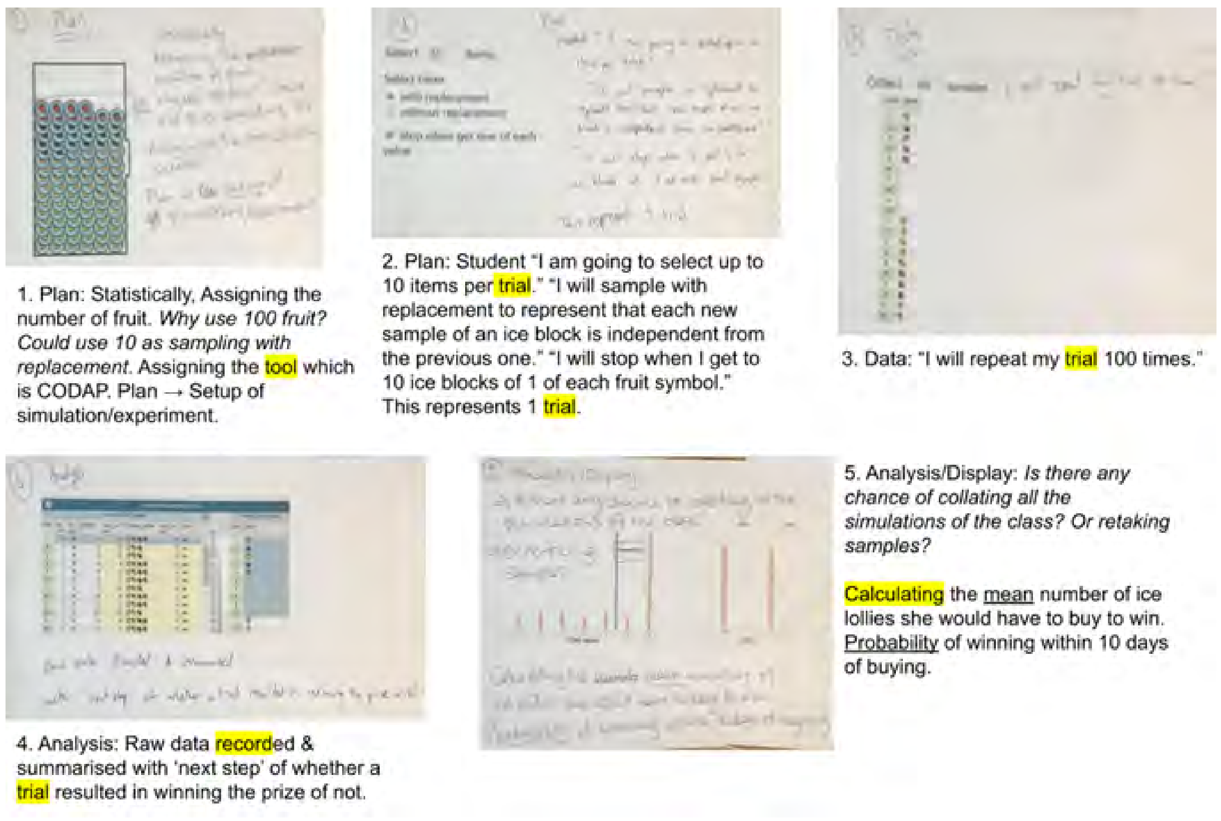
> Honestly, we didn't even think about with or without replacement, we just thought, "Oh, it's shuffled it, so it's random, so we just take one and that's our thing, and you just keep taking them", and it never occurred to us that, "Oh, you have to put it back, because that way you keep the probabilities the same."

The teachers suggested that if there had been another button in the *Simple Sampler* tool labelled "replace", then this may have triggered them to consider whether they needed to sample with or without replacement. The introduction of the new *Simple Sampler* tool seemed to activate, elicit, and support the teachers to reconsider the underpinning random sampling ideas, which were inherent and not explicitly obvious in the tools that they used in their teaching, because the new tool was more transparent about the generation of random outcomes from a probability model. Specifically, they realised that randomly sampling with replacement using objects of different types and weightings produces the same result as using a random number function to generate values that are then mapped to outcomes. Hence, their thinking about how random outcomes were generated from a probability model was reconciled by a blending of prior tool and new tool perspectives.

7.5.2 Prior task-frame thinking

I observed that when the task introduced a sequence of five modelling actions for conducting a probability simulation, that some of the statistical and computational ideas embedded in these actions were not familiar to the teachers. Because their thinking about probability simulations appeared to be associated with familiar assessment task structures, I described the theme as *prior task-frame thinking.* An example of prior-task frame thinking was related to the use and description of a probability model for simulations.

In Phase 3 of the task, the teachers were given five screenshots that each captured one of the five modelling actions needed to complete a probability simulation-based investigation for the *Fruity freezes* task. Figure 7.9 shows the screenshots provided and the order and descriptions that were written by Ella and Yin.



1. Plan: Statistically, Assigning the number of fruit. *Why use 100 fruit? Could use 10 as sampling with replacement.* Assigning the tool which is CODAP. Plan → Setup of simulation/experiment.

2. Plan: Student "I am going to select up to 10 items per trial." "I will sample with replacement to represent that each new sample of an ice block is independent from the previous one." "I will stop when I get to 10 ice blocks of 1 of each fruit symbol." This represents 1 trial.

3. Data: "I will repeat my trial 100 times."

4. Analysis: Raw data recorded & summarised with 'next step' of whether a trial resulted in winning the prize of not.

5. Analysis/Display: *Is there any chance of collating all the simulations of the class? Or retaking samples?*

Calculating the mean number of ice lollies she would have to buy to win. Probability of winning within 10 days of buying.

Figure 7.9: The annotated screenshots created by Ella and Yin, showing their names/labels and descriptions for each step

During Phase 2, Ella and Yin referred to the acronym TTRC (Tool, Trial, Record, Calculate) when they discussed how they would teach a task like *Fruity freezes*. Figure 7.9 highlights where these teachers used the words "tool", "trial", "record" and "calculate" when describing each screenshot. Ella and Yin also attempted to use part of the acronym PPDAC (Problem, Plan, Data, Analysis, Conclusion), also known as the statistical enquiry cycle. Hence Steps 1 and 2 are labelled "plan" by the teachers, Step 3 is labelled "data", and Steps 4 and 5 are labelled "analysis".

Ella and Yin, similar to two other pairs of teachers, did not use modelling language in their discussions about the steps displayed in the screenshots. Instead, they framed the probability simulation in terms of the *tool* being used, what constituted a *trial* and what needed to be *calculated*. Ella and Yin also did not use the word "assumptions" when describing the probability model for the situation described in the *Fruity freezes* task, although they did refer to using sampling with replacement "to represent that each new sample of an ice block is independent from the previous one." Overall, it appeared that the teachers were used to framing probability simulations in terms of TTRC, rather than through the lens of a modelling perspective.

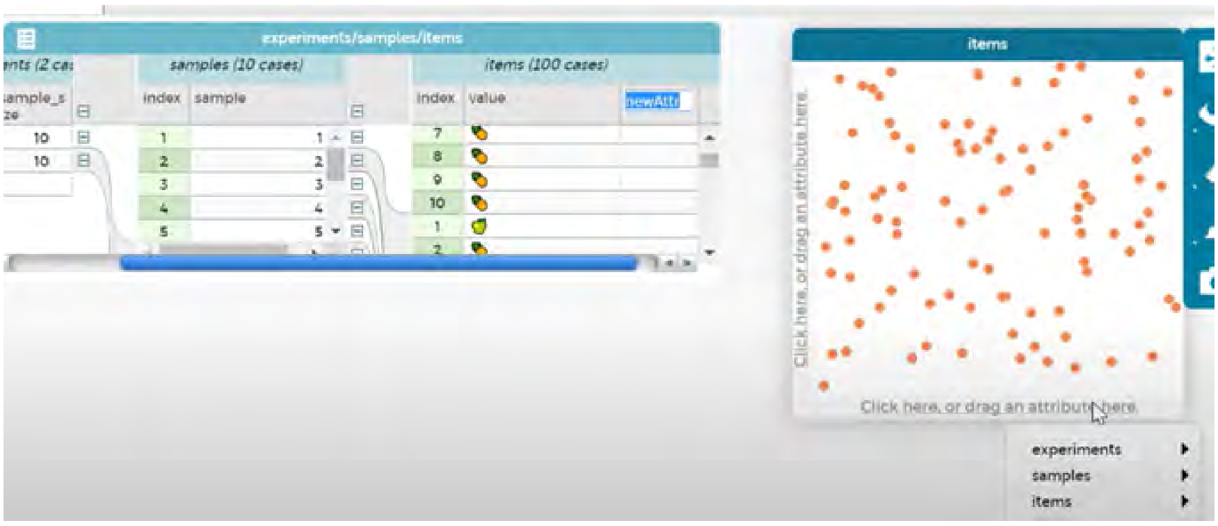
7.5.3 Tool-stimulated thinking

I observed that when the new computational tools were used to analyse data generated from a probability model, the teachers did not immediately recognise the need for the user to initiate computer-automated actions. Because it appeared that investigating the same modelling actions with different computational tools helped the teachers to develop new thinking practices, I described the theme as *tool-stimulated thinking.* An example of tool-stimulated thinking was related to the set of connected functions used to calculate the probability of winning a prize.

In Phase 2, the teachers were given two links to different pre-prepared *CODAP* files: the first link provided the model for the simulation partially set up using the adapted sampler plugin and the second link provided the table of results with the different variables needed to calculate the probability of winning a prize. Erika and Laina used the first pre-prepared *CODAP* file to generate data and immediately recognised that the model was not set up correctly. After changing the options for the sampler tool, so that each trial stopped when a full set of symbols was obtained, the teachers' attention moved to finding the number of values (symbols) generated per trial from the data produced. Their expectation was that the "number of ice blocks bought" per trial would be provided by the tool at the same time as generating the data. After the teachers realised the "number of ice blocks bought" per trial was not automatically produced by *CODAP*, Erika stated:

> How do we get the number? Why don't we do a graph or a table? We're recording on a table, we should be able to!

Figure 7.10 shows how Erika and Laina attempted to create a graph to display the "number of ice blocks bought" per trial and how they attempted to create a new variable at the case level that counted the number of values for each trial. They were unsuccessful, however, in both attempts as they did not know how to work with the hierarchal data presented.

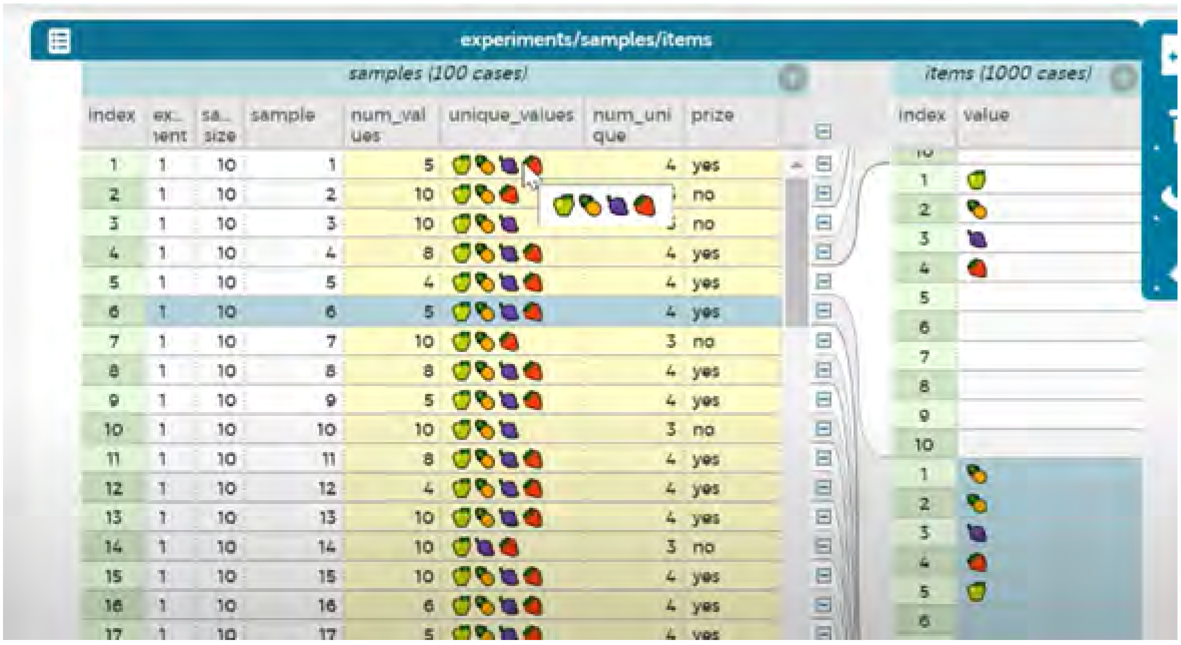

experiments/samples/items

sample_size		index	sample		index	value	newAttr
10		1	1		7	🍊	
10		2	2		8	🍊	
		3	3		9	🍊	
		4	4		10	🍊	
		5	5		1	🍏	
					2	🍊	

ents (2 cas
samples (10 cases)
items (100 cases)
items
Click here, or drag an attribute here.
Click here, or drag an attribute here.
experiments
samples
items

Figure 7.10: A screenshot of Erika and Laina attempting to use CODAP to find the number of values generated for each trial

When Erika and Laina interacted with the second pre-prepared *CODAP* file, the visual linking of a row in the parent table to the case-level values of the child table appeared to help them start to understand the hierarchal structure of the data. Figure 7.11 shows that after Laina clicked on row 6 in the “samples” parent table, the five rows in the “items” child table were highlighted.


experiments/samples/items

samples (100 cases)

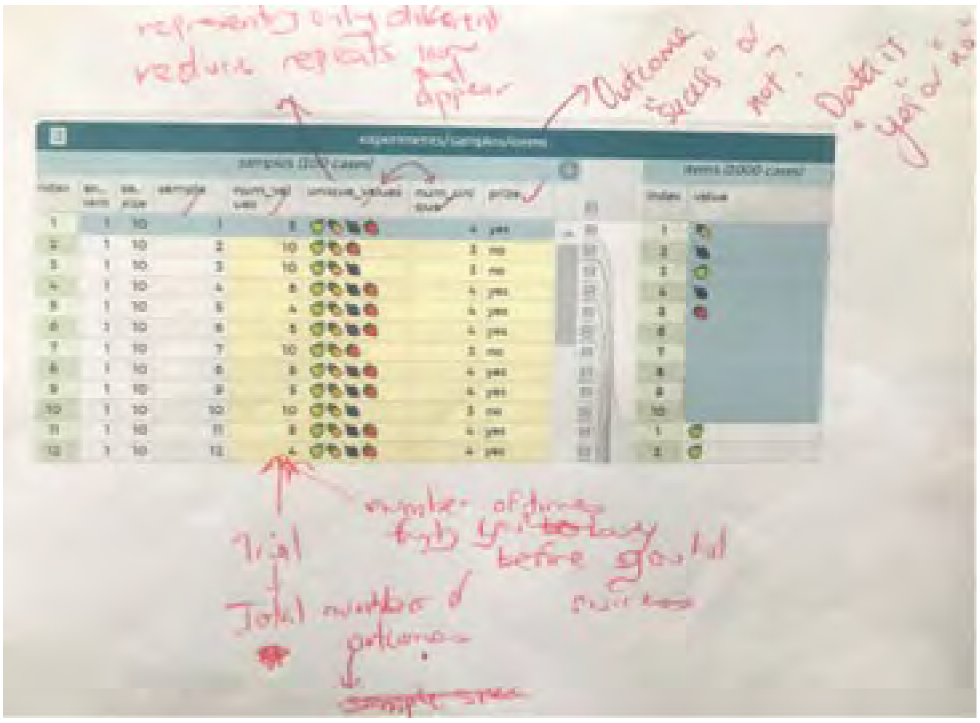
index	ex...ent	sa... size	sample	num_values	unique_values	num_unique	prize
1	1	10	1	5	🍏🍊🍆🍓	4	yes
2	1	10	2	10	🍏🍊🍓	[illegible]	no
3	1	10	3	10	🍏🍊🍆	[illegible]	no
4	1	10	4	8	🍏🍊🍆🍓	4	yes
5	1	10	5	4	🍏🍊🍆🍓	4	yes
6	1	10	6	5	🍏🍊🍆🍓	4	yes
7	1	10	7	10	🍏🍊🍓	3	no
8	1	10	8	8	🍏🍊🍆🍓	4	yes
9	1	10	9	5	🍏🍊🍆🍓	4	yes
10	1	10	10	10	🍏🍊🍆	3	no
11	1	10	11	8	🍏🍊🍆🍓	4	yes
12	1	10	12	4	🍏🍊🍆🍓	4	yes
13	1	10	13	10	🍏🍊🍆🍓	4	yes
14	1	10	14	10	🍏🍆🍓	3	no
15	1	10	15	10	🍏🍊🍆🍓	4	yes
16	1	10	16	6	🍏🍊🍆🍓	4	yes
17	1	10	17	5	🍏🍊🍆🍓	4	yes

items (1000 cases)

index	value
10	
1	🍏
2	🍊
3	🍆
4	🍓
5	
6	
7	
8	
9	
10	
1	🍊
2	🍊
3	🍆
4	🍓
5	🍏
6	

Figure 7.11: A screenshot showing how clicking a row in a parent table in CODAP highlights the relevant rows in the child table

The visual representation of the hierarchal data structure then appeared to support the teachers to understand how the summary variables `num_values` , `unique_values`, `num_unique` and `prize` were calculated, in particular that the computer needed to check how many of the symbols were different (`num_unique`) in order to determine if a prize was won. The teachers' deepening understanding of how the variables `unique_values`, `num_unique` and `prize` were linked was demonstrated during Phase 3. Figure 7.12 provides the teachers' annotations to the relevant screenshot for the modelling step.



represents only different
reduce repeats icons that appear
Outcome "success" or not?
Data is "yes" or "no"
experiments/samples/items
samples (100 cases)
items (1000 cases)
number of times fruit until before goal hit
Trial
Total number of outcomes
~~sample size~~

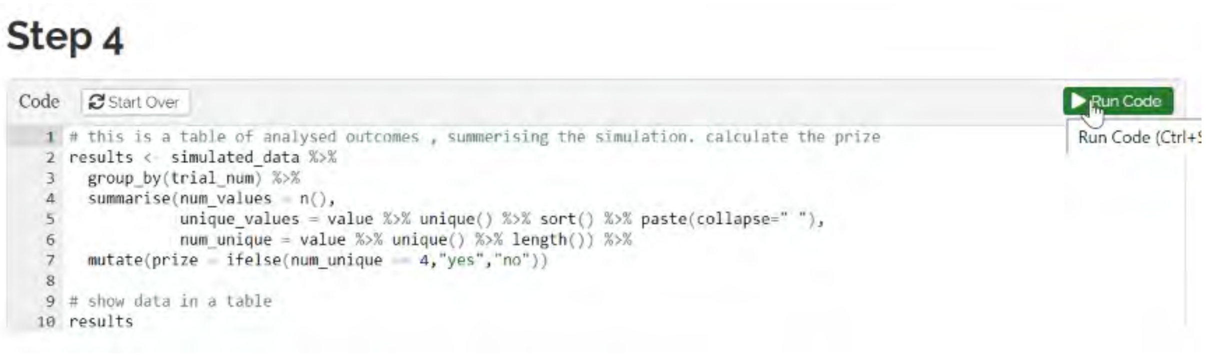
index	[illegible]	[illegible]	sample	num_values	unique_values	num_unique	prize
1	1	10	1	5		4	yes
2	1	10	2	10		3	no
3	1	10	3	10		3	no
4	1	10	4	6		4	yes
5	1	10	5	4		4	yes
6	1	10	6	5		4	yes
7	1	10	7	10		3	no
8	1	10	8	6		4	yes
9	1	10	9	5		4	yes
10	1	10	10	10		3	no
11	1	10	11	6		4	yes
12	1	10	12	4		4	yes

Figure 7.12: The annotated screenshot created by Erika and Laina for the modelling step related to the calculation of the variables `num_values` , `unique_values`, `num_unique` and `prize`

Figure 7.12 shows that the teachers drew an arrow linking `unique_values` and `num_unique` together, and their annotation for `unique_values` states, "represents only different icons that appear, reduce repeats." It did take some time for Erika and Laina to understand why the different variables were needed, as they had to put themselves in the position of a computer not a human, which Erika explained as:

> Yeah, I think that's what we're missing computationally is, all of this data, we don't know how the computer is thinking, we're still thinking like humans.

In Phase 4, Erika and Laina were presented with *R* code that matched the modelling step from the screenshot shown in Figure 7.12. Figure 7.13 shows the *R* code provided and the comments written by the teachers on lines 1 and 9.



```r
 1 # this is a table of analysed outcomes , summerising the simulation. calculate the prize
 2 results <- simulated_data %>%
 3   group_by(trial_num) %>%
 4   summarise(num_values = n(),
 5             unique_values = value %>% unique() %>% sort() %>% paste(collapse=" "),
 6             num_unique = value %>% unique() %>% length()) %>%
 7   mutate(prize = ifelse(num_unique == 4,"yes","no"))
 8
 9 # show data in a table
10 results
```

Figure 7.13: The R code provided for Step 4 of Phase 4

Erika and Laina, similar to the other pairs of teachers, were able to match the lines of code to the different variables shown on the *CODAP* table. For example, Laina said, "group by trial number, yep it's going to group it", before pointing to the `num_values` , `unique_values`, `num_unique` and `prize` variables in the table from the relevant screenshot and matching each of these to the lines 4, 5, 6 and 7 of the code. After running the code and viewing the output, Erika exclaimed:

> And you've had to use some functions to get to this, there's something really complicated going on there computationally!

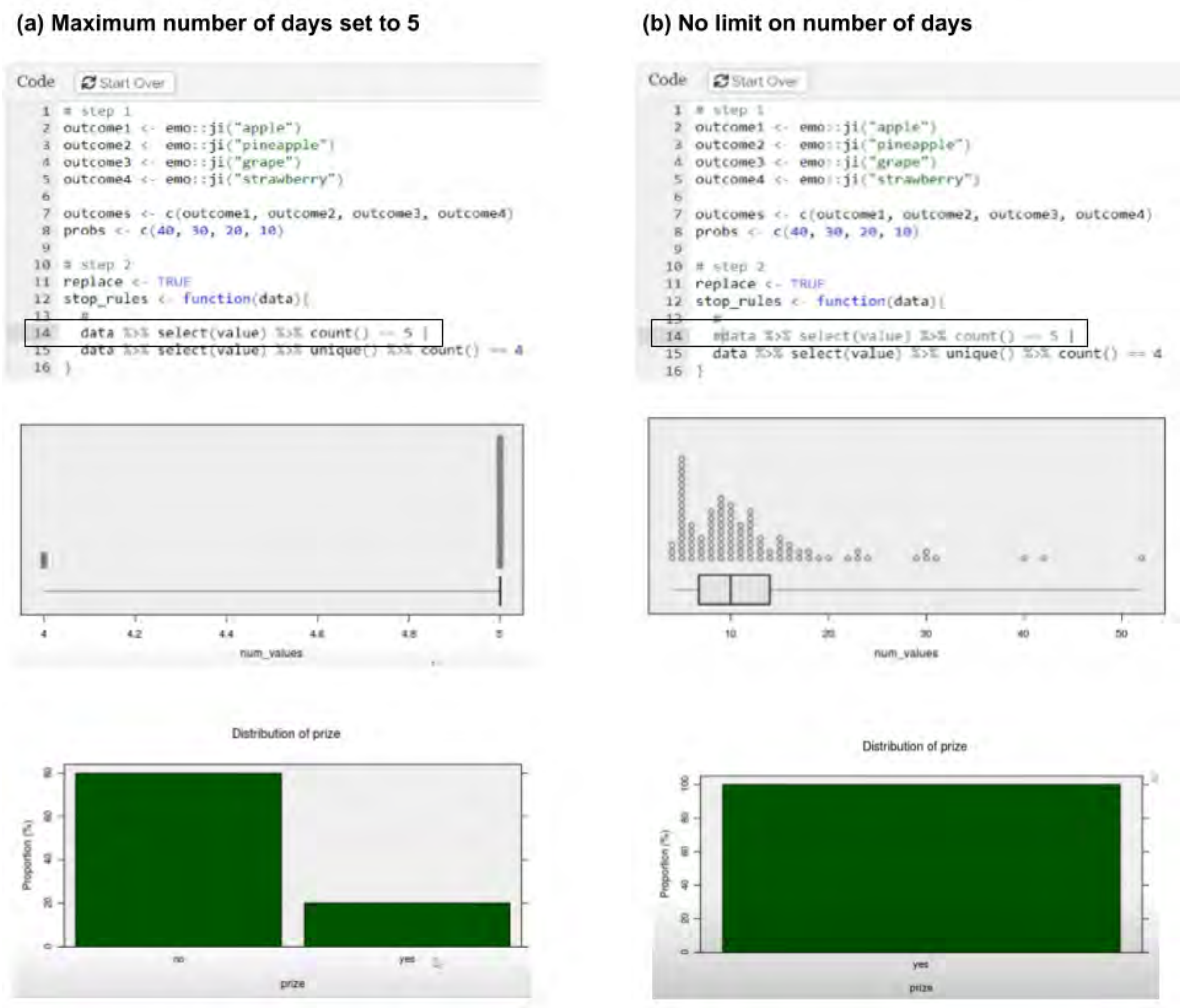
The different tools seemed to stimulate the teachers' thinking about the calculation of the probability of a prize needing computer-automated actions and a set of connected functions. They recognised that the nature of these computations was quite complex for the *Fruity freezes* task.

7.5.4 Task-stimulated thinking

I observed that when the task prompted teachers to change the probability model and to reason with the data generated, the teachers appeared to utilise thinking practices related to probability modelling. Because the task instructions that stimulated "model tinkering" were not familiar to the teachers, I described the theme as *task-stimulated thinking*. An example of task-stimulated thinking was when the teachers were asked to explore "what if?" scenarios related to the *Fruity freezes* task.

In Phase 6 of the task, three "what if?" questions were initially given to the teachers to explore. Similar to the other pairs of teachers, Naomi and Umika quickly identified and adapted the line of code that determined the maximum number of days that Grace buys ice blocks. Figure 7.14

provides two examples of their modified code, with line 14 highlighted by a box, and the plots produced when running each set of code.



(a)

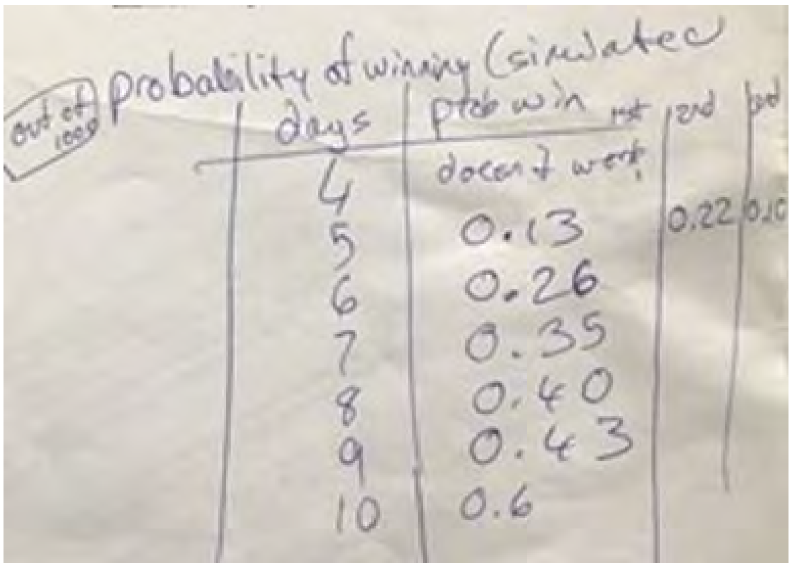
```
1  # step 1
2  outcome1 <- emo::ji("apple")
3  outcome2 <- emo::ji("pineapple")
4  outcome3 <- emo::ji("grape")
5  outcome4 <- emo::ji("strawberry")
6
7  outcomes <- c(outcome1, outcome2, outcome3, outcome4)
8  probs <- c(40, 30, 20, 10)
9
10 # step 2
11 replace <- TRUE
12 stop_rules <- function(data){
13   #
14   data %>% select(value) %>% count() == 5 |
15   data %>% select(value) %>% unique() %>% count() == 4
16 }
```

(b)

```
1  # step 1
2  outcome1 <- emo::ji("apple")
3  outcome2 <- emo::ji("pineapple")
4  outcome3 <- emo::ji("grape")
5  outcome4 <- emo::ji("strawberry")
6
7  outcomes <- c(outcome1, outcome2, outcome3, outcome4)
8  probs <- c(40, 30, 20, 10)
9
10 # step 2
11 replace <- TRUE
12 stop_rules <- function(data){
13   #
14   #data %>% select(value) %>% count() == 5 |
15   data %>% select(value) %>% unique() %>% count() == 4
16 }
```

Figure 7.14: Two examples of the code modified by Naomi and Umika, with line 14 highlighted by a box and the plots produced when running each set of code

Both Naomi and Umika articulated the change to the probability model and anticipated the plots that were produced by their code modifications. For example, when Umika used a hash symbol (#) to “comment out” line 14, she stated, “then that condition gets obsolete”, referring to one of the “stopping” conditions of the probability model. Umika also appeared to expect the plots that were produced from the modified code shown in Figure 7.14b, remarking, “that’s correct, she’ll keep on trying and trying until she gets a full set.” Naomi and Umika continued to explore how the probability of winning a prize changed, as the maximum number of days that the cards were collected increased. Figure 7.15 provides the hand-written record of their exploration.



out of 100

probability of winning (simulated

days	prob win 1st	2nd	3rd
4	doesn't work!		
5	0.13	0.22	0.10
6	0.26		
7	0.35		
8	0.40		
9	0.43		
10	0.6		

Figure 7.15: The hand-written record of Naomi and Umika's exploration for the "what if?" questions

Using the results recorded in Figure 7.15, Naomi and Umika observed that at first the probability of winning "jumped up" but then the "jumps" got smaller. Both teachers also discussed that they wanted to be able to write code to automate the generation of the "probabilities of winning" as the number of maximum days increased.

The exploration of the "what if?" scenarios, in particular the quick actions of changing the model and viewing the data produced, led Naomi to wonder about chance variation. Naomi and Umika kept the maximum number of days that the cards were collected fixed at five, and then examined the variation between the proportions calculated for winning a prize across three simulations. They observed that the simulated "probability of winning" varied (13%, 22%, 10%). Based on these results, Naomi correctly stated, "there's a lot of variation with the simulations, you're not going to get a very good estimate with 100 trials." Hence their thinking about probability modelling appeared to be stimulated by "what if?" scenarios posed by the task.

7.6 Reflection

One objective of my research was to test and refine a design framework to introduce teachers to code-driven tools for statistical modelling. Another objective of my research was to explore the statistical and computational thinking practices that might emerge when teachers were exposed to new GUI-driven and code-driven tools for teaching probability simulations. I found that the teachers were able to use different computational tools to engage with probability simulations and that the task and tools seemed to support them to successfully modify probability models and interpret features of the data generated (cf. Pfannkuch & Budgett, 2016). I now discuss the teachers' emergent thinking practices and make tentative links between these results and my task design framework.

Emergent thinking practices

One of the motivations for using the *Fruity freezes* situation was its familiarity to the teachers in my study. However, when the teachers were introduced to using GUI-driven tools to model the *Fruity freezes* situation, they did not always transfer their prior statistical understandings and did not immediately recognise the need for the user to initiate some computer-automated actions. Similar to Nilsson's (2014) research, where students explored probability models by selecting physical items from a bag, the teachers did not initially realise the need to sample with replacement when they interacted with the first GUI-driven tool, the *Simple Sampler* tool. They either assumed the tool would generate what was needed or admitted to not thinking about the sampling situation. Additionally, the *Fruity freezes* situation is quite complex to model using automation within computational tools, due to the stopping condition(s) for a trial. When the teachers were introduced to new GUI-driven tools and asked to think about and describe what was happening computationally in Phases 2 and 3 of the task (cf. Wild et al., 2017), they began to appreciate the need for computers to be "told what to do."

Although probability models are central to probability simulations, most of the teachers did not refer to the word model in their discussions throughout the task, in direct contrast to the "cereal box" example provided by Garfield et al. (2012). An explanation for the lack of reference to a probability model could be the fact that the national exemplar task (*Fruity freezes*) did not use the word model. My finding appears to give weight to the recommendation by Biehler and Schulte (2017) that modelling needs to be added as a step to the PPDAC cycle. Even though probability simulations are one of the oldest statistical modelling approaches taught at high school (e.g., Simon et al., 1976; Thomas & Moore, 1980), I propose that there is still much to learn about how to support teachers to select and combine technology for teaching probability simulations. My results indicate that national assessment tasks and tools can influence teachers' perspectives on probability simulations (cf. Sánchez, 2002) and that a modelling perspective on probability and effective use of technology is needed (e.g., Kazak & Pratt, 2021). I have presented examples of how both the features of the tools and tasks appeared to stimulate teacher thinking.

Furthermore, the repeated experiences with the same modelling steps across the tools appeared to support development of computational thinking. The teachers often made use of the CODAP screenshots to support their reading of the *R* code presented in the later phases of the task, behaviours that align with the assertion by Erickson et al. (2019) that the naming of computational actions with data might help students move between GUI-driven and code-driven tools. The readability of the *R* code appeared to further support teachers to connect the lines of text to familiar computational actions and to successfully modify the code, which is consistent with the findings of Ferreira et al. (2014). The identification of modelling steps that could transfer across computational tools, and the modification of features of these tools to support teachers to make connections between tools, were important and purposeful design decisions. These design decisions align to three of the design considerations of my framework (Table 7.1): the *introduction of new knowledge* (C1), the *tools used* (C3) and the *level of computational transparency* (C4).

Task design framework

The design principles of my task design framework also explicated the construction of task phases. Using the *immerse* (P1) principle, the first phase of the task engaged teachers with context of collecting objects. All teachers discussed personal experiences with collecting, for example, supermarket promotions or cereal cards. Unlike previous tasks implemented with teachers within the larger study, contextual information was not used to guide model decisions in the task (cf. Pfannkuch, 2011), and this may be due to the complexity of accounting for human decisions within a probability modelling context.

In the second phase of the task, using the *re-familiarise* (P2) principle, the teachers were first asked to discuss how they currently teach tasks like the *Fruity freezes* problem. The familiarity of the teachers with the specific outcomes and associated probabilities for the probability model and the two different “stopping” conditions for each trial did support them to engage with the new ideas introduced later in the phase and in later phases in the task. Unlike previous tasks implemented with the teachers in the larger study, however, there were no familiar computational tools to use during this phase, and so this phase was also used to introduce and familiarise teachers with two new GUI-driven tools: the *Simple Sampler* and *CODAP*. Although there were some difficulties with the teachers learning how to use the tools, the struggle appeared to be productive as it ensured key statistical modelling ideas were met before new computational ideas were introduced.

The importance of being able to identify modelling steps was demonstrated in the third phase of the task, when the teachers were given screenshots from *CODAP* and asked to *describe* (P3) what was happening statistically and/or computationally for each step. The teachers demonstrated engagement with new computational ideas, for instance, recognising the sequence of steps and calculations needed to determine when a trial ended. The complexity of modelling the "stopping" conditions for a trial then became the focus for phases four and five of the task, and the teachers were able to *match* (P4) and *adapt* (P5) the code provided to explore both the *Fruity freezes* and *Oh what a seed* problems using a code-driven tool. Adapting the code to match the features of the *Oh what a seed* problem, without changing data contexts (C2), provided support for the teachers to integrate statistical and computational thinking.

Using the *explore* (P6) principle, the sixth and last phase of the task encouraged teachers to explore changes to the code, with the expectation that new or unexpected outputs from these adaptations would stimulate an integration of statistical and computational thinking. The teachers successfully adapted the code provided to explore the "what if?" questions provided in phase six of the task and demonstrated a growing awareness of how technology could automate more of the statistical modelling process than is currently taught for probability simulations.

For instance, all teachers expressed wanting to be able to automate the production of data to explore how the probability of winning a prize changed as the number of days ice blocks were collected increased. When the teachers were challenged to develop the rules for prizes for a new competition, they successfully integrated statistical and computational thinking to make changes to the underlying probability model and conditions for stopping a trial, facilitated by quick adaptations to their code, similar to the pair of teachers discussed in Chapter 4.

7.6.1 Summary

In this chapter, I have demonstrated how a task can be designed to support teachers to transition to, and move between, GUI-driven and code-driven tools. My task used a six-phase approach to introduce and consolidate the teachers' interactions with the different computational tools. The learning goal for the task was for teachers to use a code-driven tool and probability simulations to develop a probability model that would inform the requirements for a fictional "card collecting" competition. Combining different computational tools within the task is included in both the design principles and design considerations of my framework, as well as the goal of the larger study to adopt a data science perspective on teaching statistical modelling at the high school level.

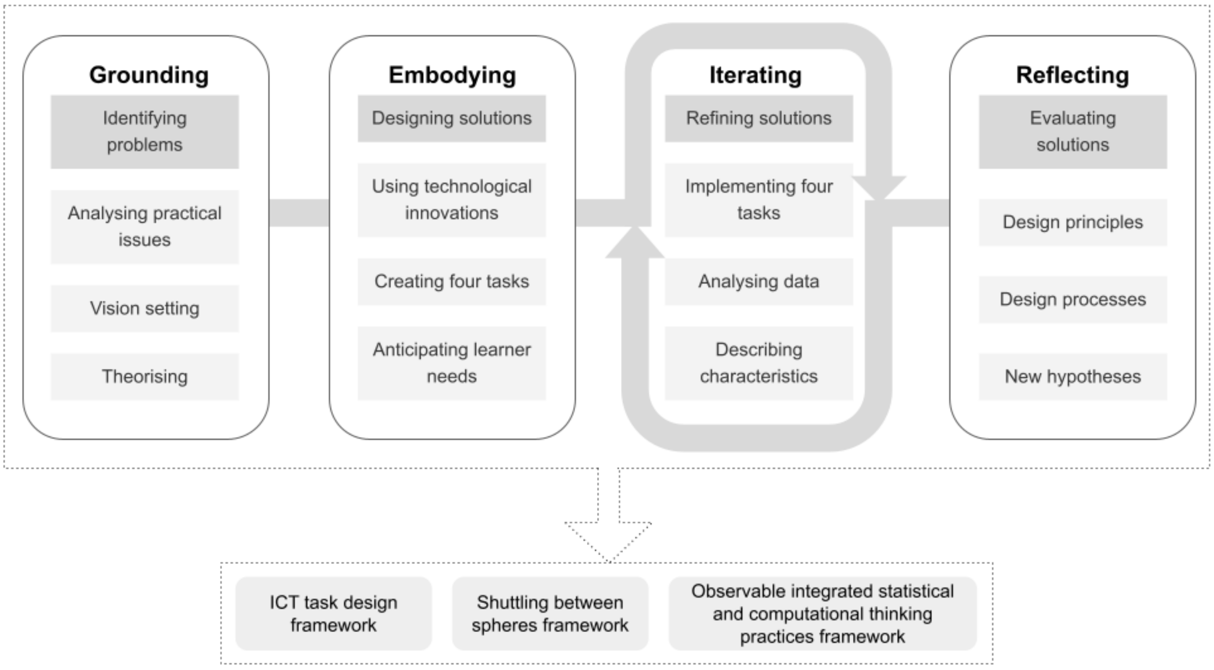
My findings indicate that there is a need to build teacher confidence to move fluently between and within digital tools and to develop their ability to choose and/or invent tools to enhance student learning. My findings also highlight how a task might facilitate teachers to play with and read code and explore “what if” probability modelling situations using code. I have elucidated how teachers’ thinking about probability simulations might draw upon previous experience and how different tools might stimulate thinking. Because a task influences how people think and act and what concepts they develop, research is recommended on specific design features of computational tools and associated tasks that support teacher and student learning for statistical modelling at the high school level.

In this chapter, I explored the introduction of code-driven tools through a simulation-based probability modelling task (Task 4) and demonstrated how the third iteration of my task design framework was modified and aligned to this task. In Chapter 8, I present the final iteration of the Introducing Code-driven Tools (ICT) task design framework and two hypothesised frameworks to support assessment of integrated statistical and computational thinking.

Chapter 8: The ICT task design framework

8.1 Introduction

Chapter 8 presents the final iteration of the Introducing Code-driven Tools (ICT) task design framework, the result of a final reflection and retrospective analysis across the entire DBR process (Figure 8.1) including the four tasks, which were presented in Chapters 4 to 7.



Grounding
Identifying problems
Analysing practical issues
Vision setting
Theorising
Embodying
Designing solutions
Using technological innovations
Creating four tasks
Anticipating learner needs
Iterating
Refining solutions
Implementing four tasks
Analysing data
Describing characteristics
Reflecting
Evaluating solutions
Design principles
Design processes
New hypotheses
ICT task design framework
Shuttling between spheres framework
Observable integrated statistical and computational thinking practices framework

Figure 8.1: DBR process model (adapted from Hoadley and Campos, 2022, p. 212)

As a designer I used my prior experience and the research literature to *ground* and *embody* the design of my tasks to introduce high school statistics teachers to code through statistical modelling. “Although design research [DBR] typically applies to the gathering of information that feeds into a creative process of design, design itself creates knowledge” (Hoadley & Campos, 2022, p. 210). Knowledge creation through design occurs through: (1) iterative reflections whereby each *iteration* is documented by describing the essential characteristics of the design solution, which in this case is the development of ICT task design framework over Chapters 4 to 7; and (2) through a final reflection across the entire DBR process (Figure 8.1).

In Section 8.2, I return to the *grounding* phase of the DBR process to review relevant literature to *theorise* about task design. In Section 8.3, I present the final iteration of my ICT task design framework. The *design principles* and *design processes* are elaborated on and discussed in Section 8.4 through a reflection across *all* phases of the DBR process that “narrates design moves, rationale, and other aspects of the design narrative” (Hoadley & Campos, 2022, p. 211) with reference to task design and other literature. Because the ICT task design framework purports to enable the integration of statistical and computational thinking, Section 8.5 presents two *new hypothesised* frameworks that seem to capture integrated statistical and computational thinking and support its assessment. The chapter concludes with a summary in Section 8.6.

8.2 Relevant theoretical perspectives on task design

To inform the final iteration of the ICT task design framework, I identified and reviewed education research literature that provided relevant theoretical perspectives on how to design *statistical modelling tasks that introduce code-driven tools*. Particular attention was given to literature about teaching statistical modelling using computational tools or introducing text-based computer programming to novices and that also substantively discussed theoretical constructs and features of tasks. Based on this review, I selected three theoretical perspectives on task design relevant to the ICT task design framework: instrumental genesis, the PRIMM model for teaching computer programming, and purpose-first programming.

8.2.1 Instrumental genesis

When designing statistical modelling tasks that introduce new computational tools, consideration is needed of how learners develop new conceptual-based and tool-based understanding simultaneously (Madden, 2021; van Dijke-Droogers et al., 2021). Instrumental genesis (Guin & Trouche, 1998), the process by which artefacts such as computational tools become instruments for one’s use, is therefore a relevant perspective to explore. As summarised by Pratt et al. (2006, p. 5), “the theory of instrumental genesis has explained how the individual gives meaning to artefacts, turning them into instruments, and accommodates himself to those instruments by inventing how they might be used” (cf. Artigue 2002; Trouche 2004). The design of a task may influence how a computational tool is used by the learner (Podworny & Biehler, 2014), and so in turn influences the instrumental genesis process.

According to Rabardel and Beguin (2005), a learner develops schema for using a particular tool while they are using it to solve a problem. Each schema represents a different thinking practice or behaviour for using the tool that connects *tool actions* with *concepts*. A pedagogical approach for teaching statistical modelling from an instrumental genesis perspective could involve providing students with a scheme for a particular tool to enable its use as an instrument. For instance, Podworny and Biehler (2014) developed a worksheet that provided screenshots of *TinkerPlots* elements arranged in the order that matched the modelling process. Text boxes were provided for each element and students were asked to complete the boxes, thus assisting them to develop their own schema. In a similar approach, van Dijke-Droogers et al. (2021) provided students with a reference document that presented key elements of a simulation-based statistical modelling approach. However, their document linked each element with a screenshot and description of the *TinkerPlots* technique, as well as a description of the related conceptual understanding, rather than students creating the descriptions themselves.

Both groups of researchers (Podworny & Biehler, 2014; van Dijke-Droogers et al., 2021) conjectured that the use of worksheets or documents to support intertwining computational techniques and statistical modelling ideas positively affected learning. Podworny & Biehler (2014) found that the completed worksheets provided a useful document for students to reflect on their approach, and van Dijke-Droogers et al. (2021) found that using the same reference document across different modelling scenarios supported conceptual development. Different computational tools place different cognitive demands on learners, and when combining different types of tools, the transitions between these tools may impact their success with modelling (Madden, 2018). *TinkerPlots* is a GUI-driven tool and learning is facilitated by a well-designed dynamic interface, where learners can point, click, and use other gestures to visualise changes to their data and models. The use of code-driven tools, where computational actions are initiated using text commands, however, places different cognitive demands on learners. Thus, theoretical perspectives related to introducing text-based computer programming to novices need to be considered.

8.2.2 The PRIMM model for teaching computer programming

The PRIMM model developed by Sentance et al. (2019) explicates a structured approach to developing a learning task for computer programming, based on the learning actions Predict, Run, Investigate, Modify, and Make. The PRIMM model provides a useful way to think about designing a sequence of learning activities where reading code is the initial focus, an approach that is consistent with longstanding research in computer science education (e.g., Van Merrienboer & Krammer, 1987). PRIMM aligns with sociocultural approaches commonly used within statistics and computer science education, where learning environments are designed that stimulate students to construct knowledge and students are encouraged to discuss their learning with others (e.g., Garfield & Ben-Zvi, 2009; Tenenberg & Knobelsdorf, 2014).

A task sequence designed using PRIMM would involve students working in small groups, being given an example of code, and then asked to *predict* what the code will produce when executed (run). After the code is *run*, the students are asked to discuss the results and compare these back to their predictions. Students then *investigate* the code and what it produces, with a focus on developing an understanding of *abstraction* and *language*. In the final phases, students *modify* the provided code to explore related problems, with any scaffolding support gradually removed, before being asked to solve a new problem that requires them to *make* their own code, drawing on the approaches introduced in earlier phases of the task sequence.

Sentance et al. (2019) described how the development of the PRIMM model was influenced by theories such as *levels of abstraction* (e.g., Perrenet et al., 2005), the *abstraction transition taxonomy* (Cutts et al., 2012), and the *block model* (Schulte, 2008). These theories seek to explain how learners develop an understanding of the syntax and semantics of code as they engage with *reading* it. Relevant pedagogical considerations include focusing beginner programmers on individual elements of the language such as keywords and statements (atoms), and on blocks of code that accomplish a specific purpose (Schulte, 2008). Learners could also be supported to transition across different levels of language in programming, such as English, computer science speak, and code (Cutts et al., 2012) and could benefit by describing and reading code aloud (e.g., Hermans et al., 2018).

Although the PRIMM model provides an important theoretical perspective on how to structure and design a task sequence to support reading and understanding code, by itself it is insufficient for designing tasks for statistical modelling. In relation to understanding code, Kaplan (2007, p.6) stated: "Good notation reveals structure; bad notation obscures it. But no notation can be clear if you haven't learned to read it and haven't learned the concepts that underlie it." Learners cannot be expected to predict what code will produce within a statistical modelling context if they have not yet had experiences with visualisations or other representations of data or models. Additionally, the goal of a statistical modelling task is not to learn computer programming. Emphasis is instead needed on how to design tasks that promote learning that connects the statistical, computational, and contextual domains, and where students are supported in purpose-driven construction and to appreciate the utility of the concepts developed (Ainley et al., 2006).

8.2.3 Purpose-first programming

Cunningham (2021) proposed that learners who are introduced to computer programming within non-computer-science subjects should be taught from a purpose-first perspective. In her research with undergraduate students from a data-oriented programming course, Cunningham found that novice programmers were often discouraged by tasks that required them to read code first, in part because code tracing can require high cognitive load (Sweller et al., 1998), and also because students perceive code reading tasks as having low value. Instead, the "purpose-first" programming approach supports programming novices to learn common code patterns associated with specific learning domains, by building on theories of programming plans (e.g., Soloway & Woolf, 1980). Specifically, students are provided with plans that achieve particular goals, for example web scraping.

Each "plan" consists of a chunk of code (the "frame") and indicates which part of the code can be changed (the "slots"). The contents of each "slot" are described using domain-specific concepts, similar to the reference documents provided by van Dijke-Droogers et al. (2021) to students when introducing *TinkerPlots* as a tool for statistical modelling. By providing all the code needed but directing learners' attention to the specific parts of the code that need to be changed to accomplish a purpose, Cunningham argued that her approach provides "glass-box" scaffolding (cf. Magana et al., 2011). In summary, the purpose-first pedagogical proposes that learners do not need to focus on reading and understanding what every line of code means. Initially, it is more important to embed the use of code within a task that is purposeful and engages students, and that students perceive as being useful (cf. Ainley et al., 2006).

8.3 Core aspects of the ICT task design framework

The purpose of the ICT task design framework is to provide practical guidance for data science teachers at the senior high school level on how to *introduce* code-driven tools through statistical modelling. The final iteration of the ICT task design framework is summarised in Figure 8.2.

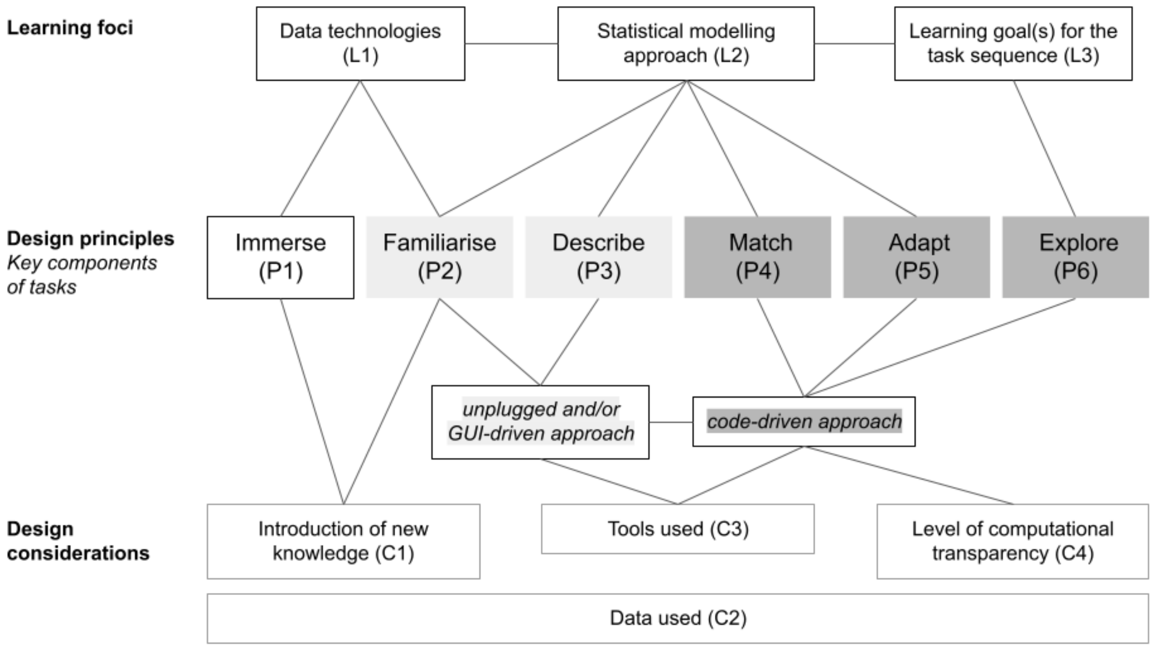


Figure 8.2: Final iteration of the ICT task design framework

The three core aspects of the ICT task design framework are: learning foci (Table 8.1), design principles (Table 8.2), and design considerations (Table 8.3).

8.3.1 Learning foci

To use the design framework to construct a task sequence that introduces a code-driven tool for statistical modelling, the task designer needs to decide what data technologies and statistical modelling approach will be used and determine the learning goal(s) (Table 8.1).

Table 8.1: Learning foci for the ICT task design framework

Learning focus	**Task designer action**
Data technologies (L1)	Select data contexts that involve data technologies
Statistical modelling approach (L2)	Identify key statistical modelling actions, representations, and associated language
Learning goal(s) for task sequence (L3)	Set learning goals that require students to create computational products

These learning foci (L1 to L3) are reviewed and refined during the design of the task sequence and finalised at the end of the construction process.

8.3.2 Design principles

The design principles (P1 to P6) are used to inform decisions about features of the learning task sequence in terms of what learners will be asked or encouraged to do, and the chronological order of these actions or experiences (Table 8.2).

Table 8.2: Design principles for the ICT task design framework

Design principle	**Task design features (what students are asked to do)**
Immerse in data (P1)	Participate in activities that promote engagement with the data context and data technologies
Familiarise with key statistical modelling actions (P2)	Carry out statistical modelling activities without using code
Describe computational aspects of statistical modelling process (P3)	Use words to describe key computational aspects of statistical modelling actions
Match statistical modelling actions to code chunks (P4)	Read and match lines/chunks of code with statistical modelling actions
Adapt code chunks with slight modifications (P5)	Identify aspects of code to change, in order to carry out statistical modelling actions
Explore “what if?” changes to code (P6)	Modify at least one aspect of provided code to produce new or unexpected outputs

Task sequences can be designed using separate phases for each design principle, or several design principles may be used within the same phase.

8.3.3 Design considerations

Alongside the design principles, the construction of the task sequence is simultaneously guided by four design considerations (C1 to C4) that inform broader decisions (Table 8.3).

Table 8.3: Design considerations for the ICT task design framework

Design consideration	**Task designer decision**
Introduction of new knowledge (C1)	Consider when and how much new knowledge is being introduced within each phase of the task sequence
Data used (C2)	Use one general data context and either one data source that provides different variables or subsets, or closely related data sources
Tools used (C3)	Connect actions and representations when learners move between different computational tools
Level of computational transparency (C4)	Select or develop features of computational tools by considering how obvious the computations performed by the tool are to learners

With respect to computational tools, the *immerse* (P1) component of the task sequence can utilise any tool, the *familiarise* (P2) and *describe* (P3) components should utilise *unplugged or GUI-driven* tools, and the *match* (P4), *adapt* (p5), and *explore* (P6) components should utilise *code-driven tools.*

8.4 Development of the ICT task design framework

The task design framework was developed a posteriori rather than a priori. Given the focus on introducing a specific tool for a specific domain of data science, the task design framework can be described as a domain-specific learning theory (diSessa & Cobb, 2004). The design framework is grounded on teaching as craft knowledge (Watson & Ohtani, 2015) and was developed using a design-based research approach. I created and implemented a sequence of four statistical modelling tasks to introduce high school statistics teachers to code-driven tools. The tasks involved four different types of statistical modelling: classification modelling, predictive modelling, randomisation tests (simulation-based inference), and probability modelling (probability simulation). The tasks were informed by three initial guidelines I developed, which in turn were based on a review of relevant literature and my years of experience designing activities for statistics teaching and assessment. These three guidelines were:

- Guideline 1: Data science tasks that introduce code-driven tools should be based on structured statistical modelling activities
- Guideline 2: Data science tasks that introduce code-driven tools should explicitly support learners to integrate statistical and computational thinking
- Guideline 3: Data science tasks that introduce code-driven tools should connect unplugged or GUI-driven tool-based modelling actions with code-driven tool-based modelling actions

After implementing the four tasks with high school statistics teachers during professional development workshops, I used four iterations of retrospective analysis to develop and refine the task design framework. Chapters 4, 5, 6 and 7 provide details about the four tasks created and their alignment to three iterations of the ICT task design framework. As the designer, I now summarise my final reflections and retrospective analysis across the entire DBR process (Figure 8.1) on the key analytical actions made at each iteration of the ICT task design framework to provide a *design narrative* (Hoadley & Campos, 2022). The narrative is linked to the literature, incorporating aspects of research and theoretical perspectives from across statistics, mathematics, and computer science education.

8.4.1 First iteration

The first iteration of the ICT task design framework was developed using the randomisation test task (Task 3, Chapter 4). An underlying assumption of the randomisation test task was that teachers were familiar with relevant statistical ideas and associated visualisations and had used unplugged activities such as shuffling cards or flipping coins to build ideas of null models and simulation-based inference before completing this task. I focused on describing how the learning in the six phases of the task was scaffolded towards using a code-driven tool for statistical modelling and what features I intentionally included in the task to support and develop statistical and computational thinking. The rich descriptions of each phase of the task and robust conversations with the wider research team (my PhD supervisors) (cf. McKenney & Reeves, 2018) led to the initial names and characterisations of six design principles: *immerse*, *re-familiarise*, *describe*, *match*, *re-use*, and *explore*. Figure 8.3 summarises the first iteration of the task design framework.

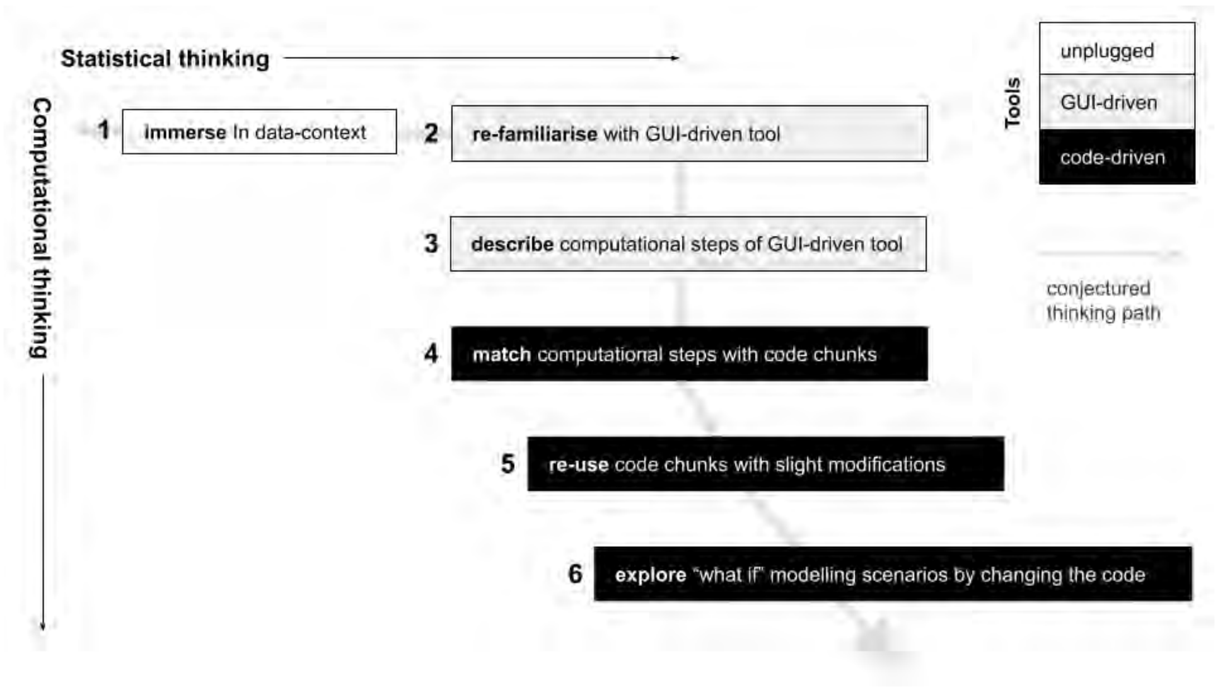

Statistical thinking
Computational thinking
1 immerse in data-context
2 re-familiarise with GUI-driven tool
3 describe computational steps of GUI-driven tool
4 match computational steps with code chunks
5 re-use code chunks with slight modifications
6 explore "what if" modelling scenarios by changing the code
Tools
unplugged
GUI-driven
code-driven
conjectured thinking path

Figure 8.3: First iteration of the task design framework

Figure 8.3 illustrates how each phase of the task was aligned with one design principle (cf. Sentance et al., 2019) and one type of tool (unplugged, GUI-driven, or code-driven). I initially conjectured that the thinking across the phases of the task should follow a sequence of statistical (Phases 1 & 2), computational (Phases 3 & 4), to an integration of the statistical and computational (Phases 5 & 6). For this iteration, I conceived of statistical and computational thinking as two dimensional, where phases that focused on statistical thinking could be illustrated as horizontal movement and phases that focused on computational thinking could be illustrated as vertical movement. Phases that focused on integrating statistical and computational thinking could be illustrated as diagonal movement, with flexibility as to the angle of the movement which represents the weighting of each type of thinking.

With a focus on developing statistical and computational thinking, Figure 8.3 explicates the design of the randomisation test task. Phases 1 and 2 aimed to support statistical thinking and focused on immersing teachers in the data context for the modelling and carrying out the randomisation test using familiar computational tools, building awareness of relevant statistical and computational aspects of the modelling task. Phases 3 and 4 aimed to support computational thinking and draw teachers' attention and thinking to the computational aspects of the modelling. These phases introduced code as a tool for articulating modelling actions and encouraged teachers' to connect familiar statistical modelling actions and visualisations with the "words" used in the code provided (cf. Clark & Paivio). Phases 5 and 6 attempted to simultaneously develop statistical and computational thinking by providing teachers with a new problem/data scenario and a modelling challenge to create test statistics, both of which required teachers to explore model production using a code-driven tool (cf. Lee et al., 2011).

Although the structured approach of using phases that aligned to design principles (cf. Sentance et al., 2019) captured core features of the randomisation task, it did not explicate other important design decisions. Three design considerations I made during the task design process included: how to introduce new knowledge (C1), what data to use (C3), and how much to reveal about the computational process using code (C4). Statistical modelling tasks that introduce code-driven tools can draw on a range of statistical, computational, context-related, data-related, and tool-related knowledge, many of which may be new to learners, and so I had to think carefully about what specific new knowledge to introduce in the task and when (C1). To minimise cognitive load (Sweller et al., 1998), I conjectured that each phase of the task should focus on the use of one tool and should introduce at most one new knowledge (cf. Wouters et al., 2008). Similarly, it appeared to be important that I had selected two data sets with the same data context (C3) (cf. Patel, 2021). Both data sets were from experiments involving the estimation of heights and the possible impact of considering values that were too high or low first, but the data sets had different properties that could be exploited to stimulate both statistical and computational thinking.

For the teachers to be able to describe and match computational steps between a GUI-driven tool and a code-driven tool, I had to make decisions about the specific code that would be used. I considered how the code could be written in a way that would make sense to teachers with little experience with using code for statistical modelling and prioritised readability (cf. Wickham, 2018). This part of the design process took a considerable amount of time and deliberation, and I finally decided to use the key modelling steps provided by the GUI-driven tool *VIT online* as the basis for code syntax decisions. As each computational step identified could be represented in many different ways using code, I identified that the task designer would need to decide the level of computational transparency (C4) that should be used across the learning task (cf. Kaplan, 2007). For example, I decided to create additional functions to the ones provided by the *infer R* package (Bray et al., 2018), as it was important to me that the teachers could see the calculation of the test statistic expressed with code and create their own test statistics, rather than use a function that hid the calculation and limited what test statistics could be used (cf. Hesterberg, 1998).

8.4.2 Second iteration

The predictive modelling task (Task 2, Chapter 5) was used to develop the second iteration of the ICT task design framework. An evaluation of the classification modelling task (Task 1, Chapter 6) confirmed that the second iteration was also applicable. As both tasks used statistical modelling approaches that were not currently assessed in senior high school statistics classes within Aotearoa New Zealand, there were some key differences between the design of these two tasks and the randomisation test task (Task 3). Unlike Task 3 where the teachers were familiar with a GUI-driven tool for conducting a randomisation test, there were no familiar tools for teachers to use to develop models that generated prediction intervals or models that classified photos using a single numeric-based decision rule. The predictive modelling task only used a code-driven tool, and the classification modelling task used "unplugged" tools alongside code-driven tools. Consequently, the design principle that was formerly described as "re-familiarise with GUI-driven tool" was re-written as "re-familiarise with statistical modelling ideas", to remove the dependency on the existence, and familiarity with, GUI-driven tools. A new design consideration was also added to the task design framework called *tools used* (C3), to draw attention to the need to consider how to combine different tools for statistical modelling (unplugged, code-driven, GUI-driven) and connect actions or representations between tools within the task (cf. Erickson et al., 2019).

The teachers' unfamiliarity with the modelling approaches required me to think differently about the design of the tasks with respect to how code-driven tools were introduced. For both tasks I used informal approaches to prediction and classification modelling, drawing on features of several MEA tasks from the CATALST project (Zieffler et al., 2019) and relevant principles of informal inferential reasoning (Zieffler et al., 2008). For the predictive modelling task (Task 2), teachers were asked to develop the error component of their model based on a visual assessment of a scatter plot rather than using formulae. Similarly, for the classification modelling task (Task 1), teachers were asked to develop a simple decision rule based on reasoning with visualisations of numeric distributions rather than using a formal algorithm. Reflections on the nature of the statistical modelling approach led me to identify that although the first three tasks had differences, a commonality was that all three tasks culminated in *producing* or creating "new" models (cf. Lesh et al., 2000), and that the learning goal was implicit in my task design decisions. These reflections on the similarities and differences in task design decisions led to the addition of two additional components to the task design framework: statistical modelling approach (L2); and learning goals for the task (L3).

Another key difference was the source of the data used for the tasks. For the randomisation test, the data was supplied as a rectangular data set, a CSV (Comma Separated Values) file, where each row was a different participant, and each column was a different variable. However, for the predictive modelling task, the data was sourced dynamically from an API (Application Programming Interface) and for the classification modelling task, the data was sourced by processing the digital image data of grayscale photos. For both the prediction and classification modelling tasks, I had to consider how to introduce and familiarise teachers with these new sources and structures of data and identified that the *immerse* phase of the task needed to support teachers to understand both the data context *and* associated data technologies (Wilkerson & Laina, 2018). These considerations resulted in use of specific task design features such as *tinker questions* (see Chapter 5, Section 2.4). Looking back across Tasks 1 to 3, I also realised that this *tinkering* approach of changing something and seeing what happens was the driver for later phases of all the tasks, and so the principle, which was formerly described as "*re-use* code chunks with slight modifications", was re-written as "*adapt* code chunks with slight modifications."

8.4.3 Third iteration

The probability modelling task (Task 4, Chapter 7) was the last task implemented with teachers and was used to develop the third iteration of the ICT task design framework. The probability modelling task was designed after the results of the first three tasks were reviewed and the first iteration of the design framework was completed. Only small changes were made to two of the design considerations, which indicates a high level of reusability. In the previous iterations of the framework, the *data used* (C2) design consideration referred to "sets of data". However, a review of the use of data across all four tasks highlighted that the data could be sourced dynamically, for example from APIs (Application Programming Interfaces), as well as from probability models. Therefore, the *data used* (C2) design consideration was changed to refer to "sources of data." The *level of computational transparency* (C4) design consideration in previous iterations of the design framework did not explicate the potential need to select or develop features of computational tools (cf. Biehler, 1997a). This directive was added, as I needed to modify the interface of the GUI-driven tool (CODAP) and develop new functions for a code-driven tool, to support teachers to move between multiple computational tools in the same task sequence (cf. Madden, 2018).

Although the changes to the task design framework were minimal, I found the process of creating the probability modelling task challenging. High school statistics teachers in Aotearoa New

Zealand are not expected to use computational tools for assessing simulation-based probability modelling, and existing tasks for teaching and assessing simulation-based probability modelling do not use *modelling language*. Additionally, the *Fruity freezes* task that I based Task 4 on involved a relatively complex probability model with respect to automation with computational tools. To scaffold teachers' learning towards using code-driven tools to carry out probability simulations, therefore, required greater attention to the co-development of concepts and tool knowledge (cf. Doerr & Pratt, 2008). Consequently, I created a new GUI-driven tool, called the *Simple sampler* (see Figure 7.1, Chapter 7) for the specific purpose of highlighting key modelling actions and associating these with the necessary modelling language (cf. Son et al., 2021). In this way, the third iteration of the ICT task design framework captured my considerations and reciprocally strengthened the components of the framework that emphasise *computational transparency* (C4) and connecting actions or representations between tools within the task (C3).

8.4.4 Final iteration

The fourth and final iteration of the task design framework involved re-examining all four tasks for their alignment to the task design framework and considering integrated statistical and computational thinking practices. The main changes made to the framework in this iteration were: the addition of *data technologies* (L1) and the visual presentation of the framework (Figure 8.2).

In earlier iterations of the task design framework, the *immerse* phases of the tasks were aligned to activities that familiarised the teachers with the data context, which included any associated data technologies. Based on further analysis and reflection across all four tasks and consideration of my proposed frameworks for integrated statistical and computational thinking (see Section 8.5), I identified that there were two different uses of data within the tasks. The *data used* (C2) needs to provide one general data context for the task and be either one data source that provides different variables or subsets, or two closely related data sources (cf. Lee et al., 2016). In the predictive modelling and classification modelling tasks (Task 1 and 2), however, the use of *data technologies* such as APIs served a greater purpose than the development of statistical modelling ideas. The use of data technologies also supported thinking about the *computational nature* of the data and supported a broader data science goal of building greater awareness of nature and diversity of data (cf. Gould, 2010). This distinction between *data context* and *data technologies* appeared important, and so a third learning focus was added to the ICT task design framework called *data technologies* (L1). The learning foci aspect of the task design framework was also strengthened by explicating the actions that the task designer should take when creating the task.

The final iteration of the ICT task design framework is summarised in Figure 8.2. Visually, the three core aspects of *learning foci*, *design principles* and *design considerations* are presented as three "layers" of the task design process. The new visual presentation of the ICT task design framework also provides guidance as to how these three "layers" are connected to each other, by the horizontal position of the components of the framework and the addition of lines showing key connections between components (cf. Fries et al., 2021). Smaller changes included changing the description of the second design principle (P2) from "re-familiarise with statistical modelling ideas" to "familiarise with key statistical modelling actions", to acknowledge that the statistical modelling approach might be new to learners. The subtle change from "ideas" to "actions" was to further reinforce the need to identify the key statistical modelling *actions* that would be translated to *computational actions* across the task sequence (cf. Woodard & Lee, 2021). Lastly, the ICT task design framework refers to a *task sequence* rather than a *task* to acknowledge that a series of connected tasks are needed.

When comparing the final iteration of the ICT task design framework against the three selected perspectives on task design (Section 8.2), the main similarities to the PRIMM framework (Sentance et al., 2019) is the structured step by step approach in the middle layer, P1 to P6, the naming of design principles and the attention to readability of code and code language development (C4). Instrumental genesis theory (e.g., Artigue, 2002) is most closely associated with the bottom layer, namely the tools used (C3), and the connections between the tools used and learners, illustrated in the connection of C3 to the unplugged and/or GUI-driven approach and code-driven approach, which in turn are connected to their respective design principles. Purpose-first programming (Cunningham, 2021) is about introducing learners to code, without the goal of turning students into programmers. With the focus on what code can do rather than on how code works and creating code for a purpose, purpose-first programming aligns with the setting of learning goals for task sequence (L3) that require students to create computational products, which is in the top layer of the ICT task design framework.

The ICT task design framework (Figure 8.2) embodies a theory about how to design a task sequence for introducing code-driven tools through statistical modelling. My final design narrative for the reflecting phase of the DBR process (Figure 8.1) included more theorising from the grounding phase, more reflection on how I created the tasks for the embodying phase, and more analysis across the four tasks in the iterating phase, which have resulted in my final account of the *design principles* and *design processes*. However, because the underlying premise of the ICT task design framework is that tasks that introduce code-driven tools should explicitly support learners to integrate statistical and computational thinking (see Guideline 2, Section 8.4, and Figure 8.3), and there is limited research on how to recognise, define and assess it, two *new hypotheses* arose from the reflecting phase, which are presented in the next section.

8.5 Assessing integrated statistical and computational thinking

For data science students to access modern data and associated technologies, students may need to learn new computational skills (Gould, 2015; Nolan & Temple Lang, 2010; Toews, 2016). Two broad uses of computational tools include: accessing and creating data; and developing statistical products, such as models and visualisations (See Chapter 2, Section 2.2). Statistical modelling tasks that introduce code-driven tools could provide an excellent learning situation for developing and assessing integrated statistical and computational thinking (cf. Wickham, 2010), as according to Lee et al. (2011) computational thinking requires analysis, abstraction, and automation. For data science to be taught as a subject at the senior high school level, not only do curricula need to provide opportunities for students to integrate computational and statistical thinking (e.g., De Veaux et al., 2017; NASEM, 2018), but also need to give clear guidelines and examples to support the *assessment* of integrated statistical and computational thinking (Gould, 2021).

It is difficult to find examples from statistics education research literature that specifically explain how to *assess integrated* statistical and computational thinking. Woodard and Lee (2021) explored tertiary-level statistics students' actions as they used code-driven tools to explore statistical problems. To assess the students' computing actions, Woodard and Lee developed the *statistical computing framework* which comprised four categories: (1) automation of computational procedures, (2) computational thinking, (3) utilising new methods, and (4) pattern recognition and decision making. A useful feature of their framework is that they *combine* statistical and computational aspects with respect to observable actions. An example is the *automation of computational procedures* category, where Woodard and Lee specify that students should be able to "use the technology to efficiently create graphs or summaries to make sense of the data, and to perform statistical calculations and use the results to make appropriate statistical decisions, without overloading cognitive processes" (p. S146).

Computational thinking is presented as a different category from automation (cf. Lee et al., 2011) and pattern recognition, and requires students to "create a solution strategy and communicate that strategy to the computer software, demonstrate critical or abstract thinking about a concept that the technology presents, revise code that has been provided to them, to perform a new task" (p. S146).

Practical examples of assessment criteria also offer some insights into how to identify integrated statistical and computational thinking in student work. In a paper about using visualisation to teach data analysis and programming, Wickham (2010) provided two rubrics for grading student work. The first rubric was used to assess data analysis tasks and has three criteria: curiosity, scepticism, and organisation. Notably, curiosity and scepticism are also two dispositions of Wild & Pfannkuch's (1999) framework for statistical thinking. The second rubric was used to assess computer programming tasks and had three criteria that focused on the code submitted: planning, execution, and clarity. The rubrics provide guidance for assessing statistical and computational thinking independently, however, the computational tools are assumed to be code-driven ones. The assessment-related examples provided by Wickham (2010) and Woodard and Lee (2021) highlight the difficulty in trying to define integrated statistical and computational thinking. If statistical computing is viewed exclusively through the perspective of using code-driven tools, then computational thinking can be incorrectly equated with using a code-driven tool (cf. Repenning et al., 2010), with too much emphasis placed on reading or writing code. Instead, the goal for using any computational tool for statistical modelling, including code-driven ones, is "not to produce technological sophistication per se, but rather access to a mode of thought" (Toews, 2016, p. 713).

As the implementation of data science at the senior high school level may involve GUI-driven tools, guidance for assessing integrated statistical and computational thinking should be *tool agnostic*. For instance, Erickson et al. (2019) defined *data moves* as "a set of actions made possible by a broad class of emerging digital tools designed to facilitate the manipulation and analysis of large, complex datasets" (p. 2). Similar to Woodard and Lee (2021), data moves are framed in terms of *computational actions*, for example, filtering, grouping, summarising, calculating, merging/joining, and making hierarchy. What is needed to assess the integration of statistical and computational thinking, however, is a conceptual framework that supports the identifying and connecting of knowledge and skills from both the statistical and computational domains (cf. Gould et al., 2018). In Chapter 2, Section 2.2, I reviewed statistical and computational thinking frameworks and posited that it may be useful to interpret "integrated" as *connected* rather than combined. From this perspective, I hypothesised two frameworks to show how I assessed the integration of statistical and computational thinking by characterising different *connections* between the context, statistical, and computational spheres when undertaking statistical modelling (cf. Stigler & Son, 2018) — the shuttling between spheres framework, and the observable integrated statistical and computational thinking practices framework — which are now discussed.

8.5.1 Two hypothesised frameworks to support assessment of integrated statistical and computational thinking

To explain my characterisation of the integration of statistical and computational thinking, I created a framework based on the notion of shuttling between contextual, statistical, and computational spheres (Figure 8.4).

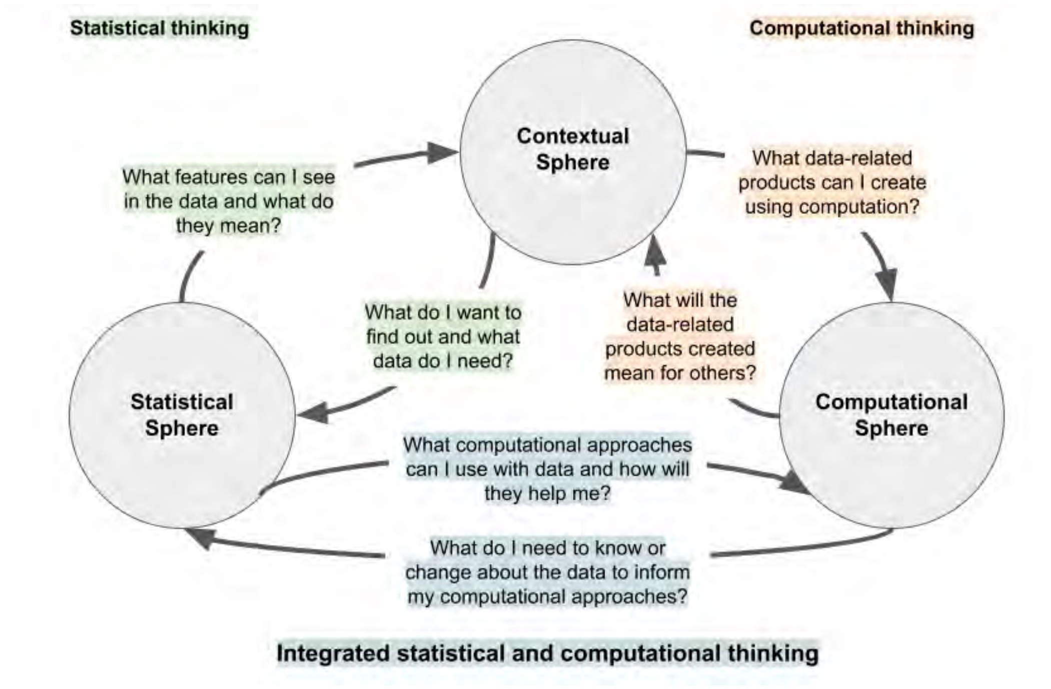

Statistical thinking
Computational thinking
Contextual Sphere
What features can I see in the data and what do they mean?
What data-related products can I create using computation?
What do I want to find out and what data do I need?
What will the data-related products created mean for others?
Statistical Sphere
Computational Sphere
What computational approaches can I use with data and how will they help me?
What do I need to know or change about the data to inform my computational approaches?
Integrated statistical and computational thinking

Figure 8.4: Shuttling between spheres framework (adapted from Wild and Pfannkuch, 1999, p. 228)

The shuttling between spheres framework draws heavily on the work of Wild and Pfannkuch (1999). The framework theorises that to identify the thinking practices of learners, one should analyse sequences of actions in terms of how learners *move between* spheres, rather than describing thinking based on observing individual *computing actions* in students' statistical work (cf. Woodard & Lee, 2021). The framework incorporates humanistic perspectives related to computational product design (e.g., Brennan & Resnick, 2012) to characterise shuttling between the contextual sphere and the computational sphere. I identified that the interplay between the *contextual* and the *computational* represented new ways of thinking about statistical modelling that were not explicitly captured by the proposed interplay between the *contextual* and *statistical* spheres (cf. Wild & Pfannkuch, 1999). Focusing on locating and connecting students thinking within and between statistical and computational spheres also aligns with emerging research involving high school students using code-driven tools for statistical modelling. Thoma et al. (2018) found that learners often framed problems as either statistical or computational, and struggled to make connections between their knowledge of the computational tool and their knowledge of statistics (cf. Woodard & Lee, 2021).

These considerations led to the six questions that have been added as annotations for each of the six shuttles on Figure 8.4, with each question deliberately written from the perspective of a learner. I found that the complexity of describing the integration of statistical and computational thinking could be partially reduced by exploring two key questions: (1) *In what ways do computational approaches support the development of statistical thinking?* (2) *In what ways does using data support the development of computational thinking?* (cf. Gelman & Nolan, 2017). In Chapters 4 to 7, I presented examples of the teachers' observed thinking practices during their engagement with four different tasks. Using Figure 8.4 and existing theoretical perspectives for statistical and computational thinking (see Chapter 2, Section 2.3), I compared and contrasted the thinking practices across the tasks to identify eight observable thinking integrated statistical and computational thinking practices (Table 8.4) that appeared to align with shuttling between the statistical and computational spheres. These thinking practices included what was observed in the teachers' actions and discussions, and what was not observed, that is, cases where a thinking practice could have been advantageous. I also considered my expectations and intentions as the task designer for the nature of thinking that could be supported or stimulated by different task features.

Table 8.4: Observable integrated statistical and computational thinking practices framework

Observable thinking practice	**Example from research tasks**
Connecting features of visualisations with statistical model components expressed computationally (OTP1)	Linking the band visualised around the fitted line to the line of code that determined the error component of the prediction model that generated prediction intervals (predictive modelling task)
Articulating a sequence of statistical modelling actions in computational terms (OTP2)	Describing the computational aspects of each modelling step using natural language, to match the computational actions captured in the screenshots from VIT online (randomisation test task)
Automating statistical modelling actions effectively using computational tools (OTP3)	Copying and adapting the given code for a probability simulation to explore the *Oh what a seed* problem situation and successfully producing a viable model for the card collecting promotion (probability modelling task)
Tinkering with statistical models productively to develop generalisations (OTP4)	Exploring new test statistics by changing aspects of the code for the model, leading to generalisations about the relationship between the features of the sample data, the test statistic, and the re-randomisation distribution (randomisation test task)
Restructuring or manipulating data for a statistical modelling purpose (OTP5)	Recognising the hierarchal nature of the data generated from the probability simulation and how it was represented in the computational tools (probability modelling task)
Developing algorithms by analysing data and selecting relevant variables (OTP6)	Developing a decision rule to classify grayscale photos as high contrast or low contrast by determining what features of the grayscale data to use for each photo and what “cut off” value to use (classification modelling task)
Obtaining data from digital sources and using data strategically to develop models (OTP7)	Sourcing data from the OMDb by using code to making queries to the API, then using the data from one query for training and the data from another query for testing (predictive modelling)
Considering uncertainty and generalisability when using models as computational products (OTP8)	Discussing how random samples of pixels were being used to create the grayscale distributions for photos and the impact of this on model developed, specifically the decision rule (classification modelling)

Each observable thinking practice in Table 8.4 connects knowledge or use of data, models, and computational tools and is illustrated using one example from the research tasks. A general characteristic for each observable thinking practice is that the learner needs to draw on aspects of analysis, abstraction, and automation, the three core aspects of computational thinking according to Lee et al. (2011). Each observable thinking practice also demonstrates the need to *shuttle between spheres*, namely the *statistical* and *computational* spheres. Consequently, the observable integrated statistical and computational thinking practices framework provides a guide for assessing integrated statistical and computational thinking, and some guidance for designing data science learning tasks. From a practical teaching perspective, it is difficult to strike the right balance between too general and too specific assessment criteria. A limitation of my Table 8.4 framework is that it is over-fitted to the teaching context of statistical modelling, and hence focuses on *connections* between the statistical and computational spheres based on *data* and *models*. Nonetheless, the observable integrated statistical and computational thinking practices framework alongside the ICT task design framework could provide useful support to data science teachers at the senior high school level to create statistical modelling tasks that introduce code-driven tools.

8.6 Summary

Knowledge creation through design is a *bricolage* (Gravemeijer, 1994), as it is practice based, drawing on all available means such as experience, other research, experimentation, trialling, learning and critical reflection. From this perspective, I have presented an explication of how to design tasks to introduce students to code through statistical modelling and a visualisation of that design in the ICT task design framework (Figure 8.2). Furthermore, I have articulated a possible way of conceiving and assessing the integration of statistical and computational thinking (Figure 8.4 and Table 8.4). Chapter 9 concludes this research project by answering the research questions, and by discussing limitations of the research and recommendations for practice and future research.

Chapter 9: Summing up and looking forward

9.1 Introduction

In Chapter 1, I explained that my motivation for this study was the realisation that I needed to rethink my task design approach when using data students interact with in their daily lives (Gould, 2015) and that the design would require thinking about how to integrate statistical and computational thinking. The lack of literature on how to create effective tasks for teaching statistics using programming and the need for me to explicate and communicate the design principles and considerations underpinning my tasks, led me to undertake this research project. Hence, within the context of teaching statistical modelling at high school, I have explored, produced, and articulated an ICT task design framework, hypothesised two Integrated Statistical and Computational Thinking (ISCT) frameworks, and provided evidence that the tasks seemed to help the teachers to integrate statistical and computational thinking.

Chapter 9 provides my final reflections on the research questions in Section 9.2 including my contributions to the research knowledge base, acknowledges the limitations of my research in Section 9.3, and discusses implications for teaching and research in Section 9.4. The chapter finishes with concluding remarks in Section 9.5.

9.2 Final reflections on research questions

The final reflections on my two research questions consider the study objectives and point out how my research has added to and confirmed existing research and provided new contributions to the statistics and data science education research knowledge base.

9.2.1 What observable thinking practices emerge as teachers, positioned as learners, engage with statistical modelling tasks that introduce code-driven tools?

The first objective for this study was to describe the observable thinking practices that emerge when teachers completed tasks that introduced code-driven tools within the teaching and learning context of statistical modelling. Supporting research questions for this objective were: *Can these observable thinking practices be characterised as integrating statistical and computational thinking?* and if so, *What features of the tasks appear to stimulate or support the integration of statistical and computational thinking?* As the teachers in the study were familiar with high school level statistics, I explored what new thinking practices emerged due to the contextual and practical computational demands of the statistical modelling tasks.

The introduction of code-driven tools through statistical modelling appeared to provide a solid basis for stimulating the integration of statistical and computational thinking. Modelling is core to both statistical and computational thinking, however, statistical models are not always viewed as *computational products* within existing statistical modelling tasks or statistical enquiry frameworks (cf. Wild & Pfannkuch, 1999). In this research, the four tasks presented statistical models as computational products that could be developed, tested, and used "in production", which appeared to support the development of computational thinking by simultaneously drawing on abstraction, automation, and analysis (Lee et al., 2011). For example, abstraction was demonstrated by the teachers' development of and generalisations about their models, automation was demonstrated by the teachers' engagement with the code representations of their models, and analysis was demonstrated by the teachers' interpretation of the features of data used to create their models or produced from their models.

Statistical thinking requires learners to understand that data are numbers with context (Cobb & Moore, 1997) and thus humanistic perspectives of model outputs are needed that account for contextual implications. Involving teachers in the creation or manipulation of data used for modelling appeared to support their humanistic thinking (cf. Lee et al., 2021), as the teachers often needed to connect features they perceived as humans, to features of the data, and then to computer extractable features (cf. Horton et al., 2021). These human-driven decisions sometimes led to uncertainty with the modelling process, and my findings include examples of the teachers grappling with the differences between how humans and computers make decisions (cf. Biehler & Fleisher, 2021). I also found that the context did at times distract the teachers from forming more general ideas about statistical models (cf. Pfannkuch, 2011; Zieffler et al., 2021), and at other times the teachers' consideration of contextual knowledge needed to be suppressed while they engaged with the underlying structure and properties of the particular statistical model being explored (cf. Cobb, 1997). Teachers were often frustrated when they knew what they wanted the computer to do *statistically* but didn't know how to make the associated modelling action happen using code (cf. Woodard & Lee, 2021).

To understand and assess the observable thinking practices of the teachers, I developed two hypothesised ISCT frameworks based on the notion of *integration* as *connection* and by extending Wild and Pfannkuch's (1999) metaphor of shuttling between spheres to include the computational sphere (Figure 8.4). The *observable integrated statistical and computational thinking practices framework* (Table 8.4) provides characterisations of eight integrated statistical and computational thinking practices illustrated with examples from the teachers' interactions with the tasks used in this research. I found that specific task design features appeared to support the teachers to integrate statistical and computational thinking. For instance, using web-based tasks that comprised a sequence of steps that carefully ordered the introduction of new statistical and computational ideas appeared to minimise cognitive load (cf. Wouters et al., 2008). The use of small chunks of code that only required small modifications allowed teachers to engage with new computational ideas such as creating scatterplots, similar to the findings of Wiedemann et al. (2020). The design and use of tinker questions also appeared to support the introduction of new computational ideas and the development of *data habits of mind* (cf. Finzer, 2013) as they provided guidance for noticing and considering selected features of computational representations or actions.

I have characterised the integration of statistical and computational thinking through observable thinking practices and identified features of the tasks that seemed to support integration, both of which contribute to a better understanding of how to recognise, design for and assess the integration of statistical and computational thinking.

9.2.2 What design principles could guide the construction of statistical modelling tasks that introduce code-driven tools?

The second objective for this study was to develop a task design framework, comprising a cohesive set of design principles and processes, to guide construction of statistical modelling tasks that introduce code-driven tools. Supporting research questions for this objective were: *How could tasks be constructed to support the introduction of new sources of data and modelling* *approaches, simultaneously with new code-driven tools?* and, *Does using familiar computational tools or modelling approaches within the same task support the introduction of new code-driven tools?* Given that teachers were familiar with existing statistical pedagogical approaches involving technology, I explored what changes to the task design process were needed to create statistical modelling tasks that introduced coding.

Two core threads through the task design principles and processes presented in the ICT task design framework (Figure 8.2) are: (1) extend the familiar into the unfamiliar (e.g., Biehler & Schulte, 2017) and; (2) use the informal before the formal (e.g., Gravemeijer, 2004). Both threads have been explored substantially in the statistics education research literature and their relevance for the construction of data science tasks has now been demonstrated in my research (cf. Gould, 2021). The statistical modelling tasks I created used teachers' familiarity with numeric data distributions, probability simulations, simple linear regression, and randomisation (permutation) tests as a basis for introducing new contextual and practical computational ideas. The tasks share similar properties to MEAs (cf. Garfield et al., 2010) and model development sequences (cf. Lee et al., 2016), by using a structured approach that combines guided discovery and instruction (cf. Grover et al., 2015). My findings also build on statistics education research that shows the benefit of using of *unplugged* activities before computational tools for statistical modelling (e.g., Chance et al., 2004; Zhang et al., 2021) and encouraging informal model-based reasoning (e.g., Makar & Rubin, 2018; Zieffler et al., 2008).

I found that the use of data technology contexts appeared to stimulate thinking about new computational approaches (cf. Hicks & Irizarry, 2018). Accessing dynamic movie ratings data, for example, provided a rich computational context to support the teachers' development of predictive modelling ideas (cf. Weiland, 2016) without distracting them from core statistical concepts (cf. Bargagliotti & Groth, 2016). Setting the statistical modelling task within a modern data technology context and structuring a task sequence towards an end goal of creating a statistical model as a computational product appeared to help teachers appreciate both the *purpose* and *utility* of their learning (cf. Ainley et al., 2006). My task design approach has similarities to the purpose-first approach (Cunningham, 2021), the structured task approach employed by the PRIMM model (Sentance et al., 2019), and the use of "computational templates" to introduce code to statistics students (e.g., Finch et al., 2021; Hardin, 2018; Grolemund & Wickham, 2014; Wickham, 2010). However, one important contribution of my research is *the explication of the task design process that led to* the creation and use of the computational templates within a statistical modelling task sequence.

The ICT task design framework I created combines and extends previous research in statistics and computer science education and provides new pedagogical theories. I have explicated the need to: identify the modelling actions or steps that could transfer across computational tools; create adaptable code chunks that represented each modelling action; consider the *computational transparency* of the modelling action when enacted by a computational tool; and connect each modelling action across the different computational tools used in the task sequence, both visually and textually. In my findings, I have presented instances where the teachers did not immediately recognise the need for the user to initiate some computer-automated actions, but when asked to think about and describe what was happening computationally (cf. Wild et al., 2017), were able to appreciate the need for computers to be "told what to do." I have reported how some teachers continued to refer to visual representations embodied in physical stimuli and GUI-driven tools to support their reading of the *R* code presented, highlighting the need to consider how well-designed GUI-driven tools can support the informal and open exploration of data and models (e.g., Ben-Zvi, 2000; Wild, 2018) before moving to code-driven tools.

I have provided examples of how the readability of the *R* code appeared to support teachers to connect computational actions with code (cf. Ferreira et al., 2014; Kaplan, 2007), strengthening my assertion that *computational transparency* is a new but important consideration in task design (cf. Hesterberg, 1998; Burr et al., 2021). In alignment with Son et al. (2021), the use of *R* code in the task appeared to be germane load (Sweller et al., 1998), potentially because the computations represented by the code were familiar to the teachers. The focus on the language of modelling, specifically on words that can connect modelling actions across different computational tools, aligns with computer science education research regarding reading and verbalising code (e.g., Hermans et al., 2018; Van Merrienboer & Krammer, 1987), and the naming of computational actions with data (cf. data moves, Erickson et al., 2019). The relationship between computational tools, task design and learners' statistical conceptions is complex (e.g., Biehler, 1997b; delMas, 1997; Doerr & Pratt, 2008) and my research contributes to a greater understanding of how learners could simultaneously develop conceptual-based and tool-based understanding (cf. Artigue, 2002).

9.3 Limitations

As this was a small exploratory study involving ten high school statistics teachers, the findings from *these learners* cannot be generalised to all high school statistics teachers. There are further limitations to the applicability of the findings of my research for the following reasons.

First, high school statistics teachers were used as participants for the study, rather than senior high school students. As computer programming is not required as part of the New Zealand school curriculum for statistics, it was not feasible to implement the tasks I had designed within senior high school statistics classrooms. Therefore, I focused my research on high school statistics teachers, and positioned them as the learners for the teaching experiments. All the teachers were familiar with senior level statistics content and therefore had a strong pre-existing knowledge base on which to incorporate new approaches to statistical modelling. Although the teacher participants had minimal to no experience with data technologies and had not used coding-based approaches for teaching statistical modelling, they were open to learning new technology and were interested in teaching data science at the senior high school level.

Second, the research involved the design and implementation of only four tasks within four full-day workshops. All these tasks were created by me and reflect my personal beliefs and knowledge about task design and were shaped by my perspective on data science education. All four tasks were highly structured and were designed for the specific purpose of introducing code-driven tools through statistical modelling. Hence, the tasks were not intended to be complete tasks for learning about statistical modelling approaches, nor do they explore important aspects of data science such as ethics, data responsibility, data ownership, data sovereignty, algorithmic bias, and related political and historical contexts of data (cf. Wilkerson & Polman, 2020). Consequently, the observations of teachers' thinking practices are constrained by the design of the tasks, but it is plausible that tasks designed in different ways could produce similar results.

Third, there are a wide range of code-driven tools and environments available for teaching data science. In this study, the computational environments were limited to web-based tasks and environments, including the use of RStudio Cloud, RMarkdown documents, and interactive tutorials created using the R package *learnr*. I did not explore the use of notebooks (cf. Fleischer et al., 2022), R scripts, or other code-driven tools such as graphics calculators (cf. Burrill, 2017). As the task designer, I made many decisions about what packages to use, what functions to use, what code syntax to use, and how to name objects. These decisions, along with the interface of the code-driven tools used, will have influenced the teachers' learning (Abbasnasab Sardareh et al., 2022) and further research is needed to determine to what extent (e.g., Myint et al., 2020).

Fourth, the analysis focused on the design of the task, and did not specifically take into account the learning environment I created as the teacher/researcher guiding the learners/teachers through the task. As I have more than 20 years' teaching statistics, significant curriculum and assessment design experience, and advanced computing skills, the learning environment I created in the teaching experiments is an important mitigating factor when interpreting my findings. Teachers were asked to reflect on the tasks and their suitability for teaching at end of each workshop, but my research did not explore the teachers using the framework to design tasks nor measure their capacity to recreate a similar learning environment when implementing the four tasks used in the research with their own students.

9.4 Implications for teaching and research

The outcomes of this study have several implications for teaching and future research. The ICT task design framework is already being used by me and other teachers within my department to develop tasks for undergraduate statistics and data science students in their first and second year at the University of Auckland. Further research is planned to collect and analyse student responses to tasks, and use these to refine the observable integrated statistical and computational thinking practices framework (Table 8.4) (cf. Woodard & Lee, 2021). Additionally, greater exploration of the impact of computational environment and software interfaces is needed. The tasks used in this research were developed during 2018 to 2019. New advances in web-based tools to support student learning need to be considered, including computational environments that combine code-driven tools alongside GUI-driven tools.

Greater professional development for statistics education is already an important priority (Bargagliotti & Franklin, 2015), and this required support extends to the implementation of data science education at the senior high school level. The teachers who participated in my research were enthusiastic about using similar data science tasks with their students but needed practical support to increase their own skills and that of their colleagues with using code-driven tools for statistical modelling. I have already shared key findings from my research with high school teachers and facilitated practical workshops using adapted versions of the research tasks, but much more is needed to equip teachers with the pedagogical content knowledge needed to implement tasks confidently (Shulman, 1986).

Future research is needed to understand how useful the ICT framework is for assisting teachers to create tasks for introducing code driven tools through statistical modelling and to learn more about how the tasks created using the ICT framework support students learning (cf. Sentance et al., 2019). A strategy to support the introduction of data science at the senior high school level could involve working with a smaller group of interested teachers to create and trial a data science curriculum suitable for Aotearoa New Zealand (cf. Begg, 2004), one that includes and promotes Matauranga Māori and Te Ao Māori approaches to data.

9.5 Concluding remarks

In this thesis, I have explored the design and implementation of four statistical modelling tasks that introduced code-driven tools. The consideration of the practical implications of implementing data science as a subject at the senior high school level led to a focus on task design, as I had identified that there were limited research-based theories or frameworks to guide teachers' construction of tasks. Although other studies within statistics and computer education have considered data science learning environments involving code-driven tools, this is the first substantial study *focused on task design* that specifically considers *how to support learners to move from non-code-driven tools towards code-driven tools* within the learning context of statistical modelling. My research can enhance the implementation of data science at the senior high school level by providing theoretical and practical guidance for creating tasks and assessing integrated statistical and computational thinking.

Appendix A - Teacher questionnaire

Questions used

- What is your name?
- What is your gender?
- How many years have you been teaching at the high school level?
- How many years have been teaching New Zealand Curriculum Level Seven or Eight Statistics (Year 12 or 13)? Include this year if applicable.
- Did you complete a major in statistics or operations research (or equivalent) as part of your tertiary study? (Yes/No)
- Did you complete a major in mathematics as part of your tertiary study? (Yes/No)
- Did you complete a major in computer science or information science (or equivalent) as part of your tertiary study? (Yes/No)

Appendix B - Discussion and reflection questions

Discussion questions

- Have you used this data context in your teaching before?
- Are you familiar with this data context?
- What key concepts do you think this part of the learning task might help build understanding of?
- How do you think your students might react to this learning task?
- What potential issues can you see with using this learning task with your students?
- How would you modify/change aspects of the learning task?
- What other knowledge or skills do you think are needed before, during or after use of this learning task?

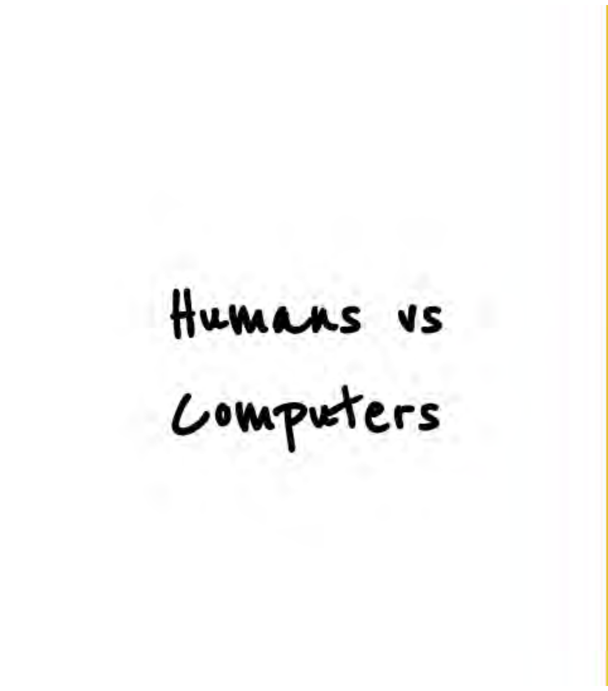
Reflection questions

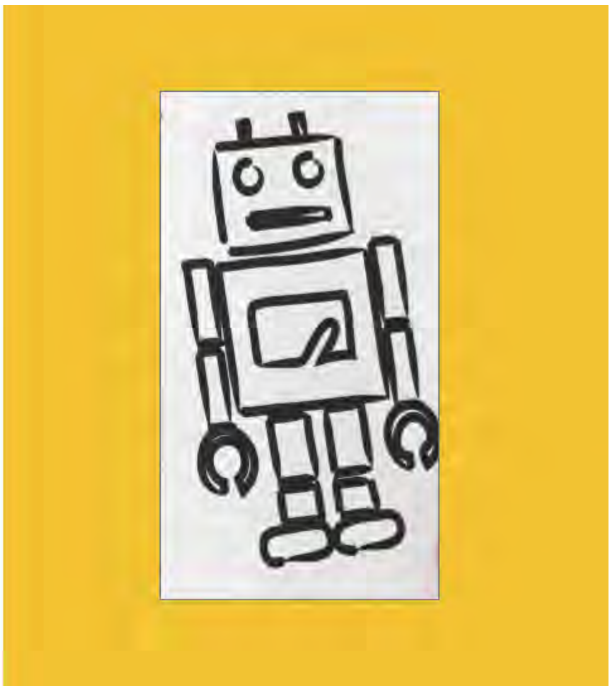
- What key concepts related to statistical thinking do you think these workshops might help build understanding of?
- Are there any concepts related to statistical thinking you feel you understand better having completed these workshops?
- What key concepts related to computational thinking do you think these workshops might help build understanding of?

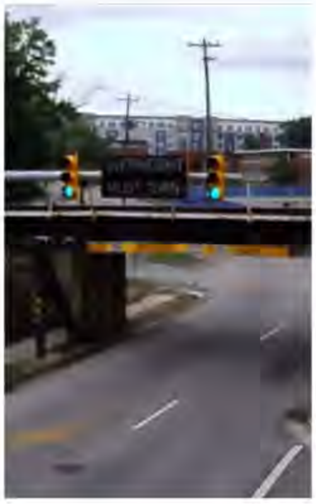
- Are there any concepts related to computational thinking you feel you understand better having completed these workshops?
- How do you think computational thinking could enhance statistical thinking?
- Is there an example of something done in the workshops that illustrates this?
- What new data-related knowledge do you think these workshops might help build understanding of?

Appendix C - Randomisation test task

Slides used during task


Humans vs Computers





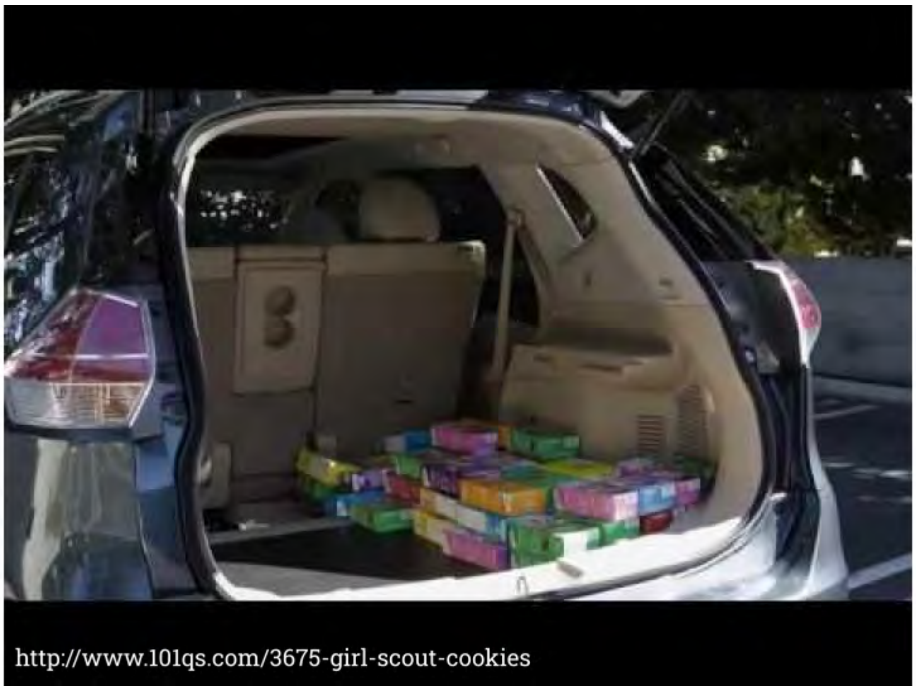
How many metres between the road and the bridge (yellow bar)?

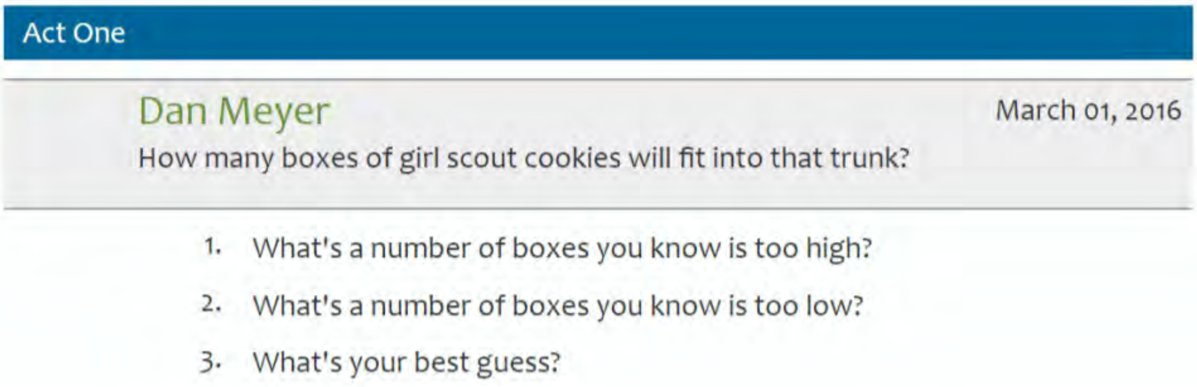
Do you think this truck will make it under the bridge?

https://www.youtube.com/watch?v=iQfSvIgIs_M

http://abc7ny.com/traffic/new-system-would-warn-truckers-about-low-overpasses/93427/

Can we influence humans' estimates of things like heights?





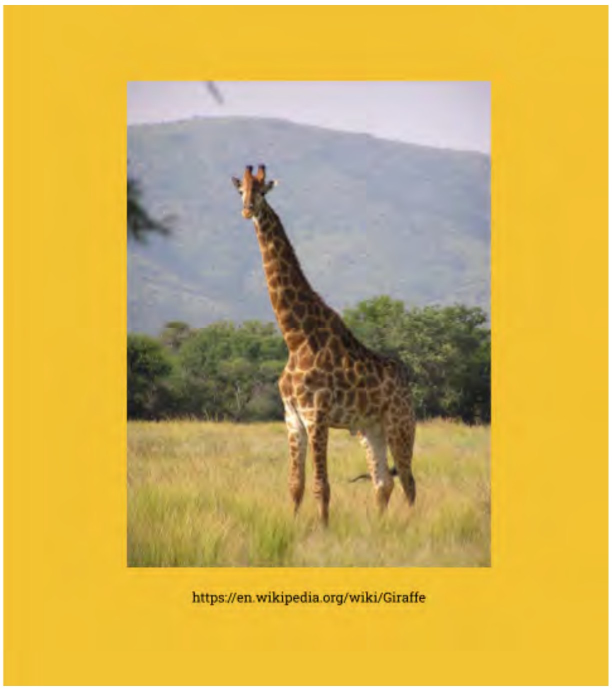
Act One

Dan Meyer March 01, 2016

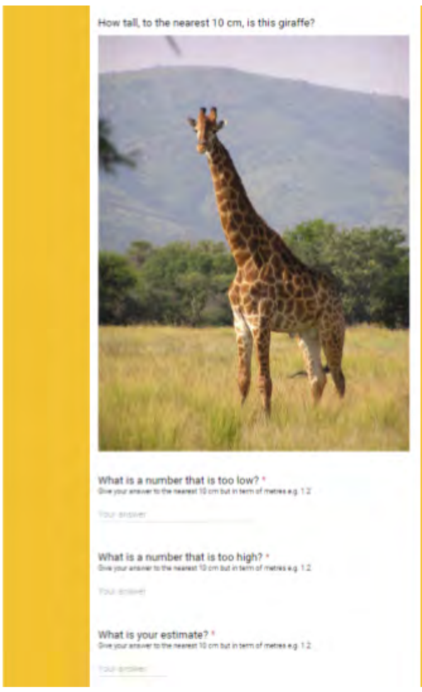
How many boxes of girl scout cookies will fit into that trunk?

1. What's a number of boxes you know is too high?
2. What's a number of boxes you know is too low?
3. What's your best guess?

How tall is this giraffe?



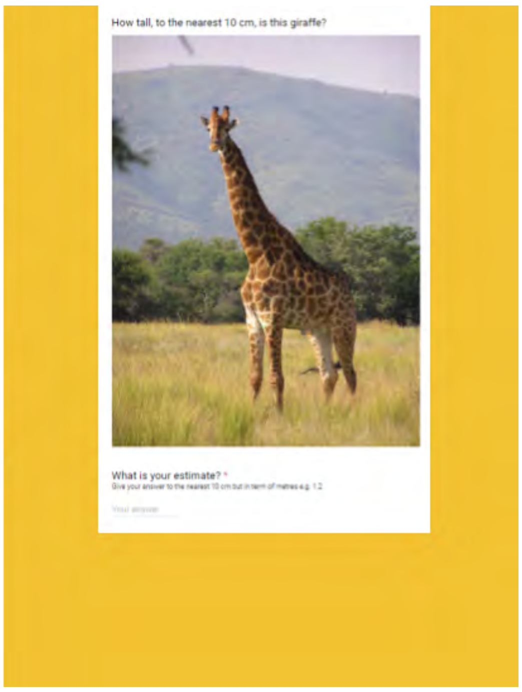
Google forms


How tall, to the nearest 10 cm, is this giraffe?

What is a number that is too low? *
Give your answer to the nearest 10 cm but in term of metres e.g. 1.2
Your answer

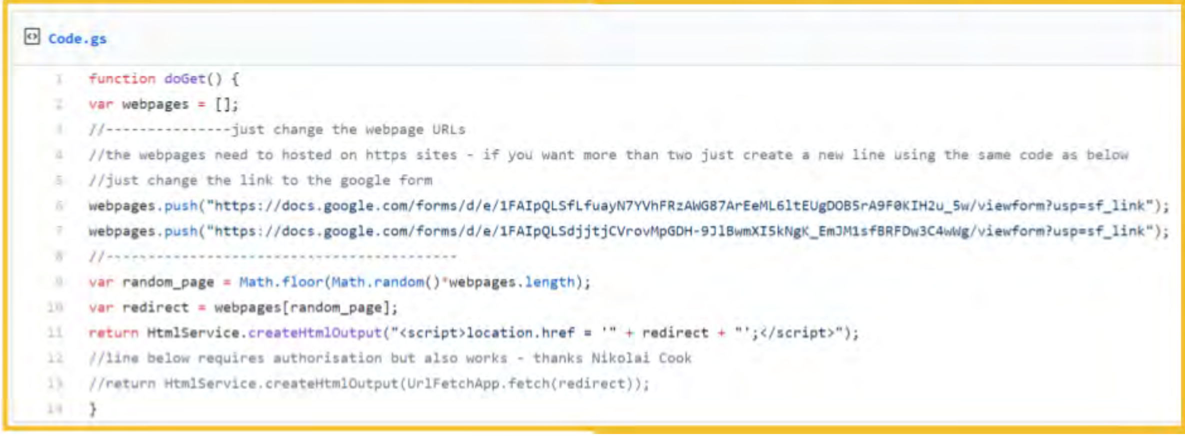
What is a number that is too high? *
Give your answer to the nearest 10 cm but in term of metres e.g. 1.2
Your answer

What is your estimate? *
Give your answer to the nearest 10 cm but in term of metres e.g. 1.2
Your answer


How tall, to the nearest 10 cm, is this giraffe?

What is your estimate? *
Give your answer to the nearest 10 cm but in term of metres e.g. 1.2
Your answer

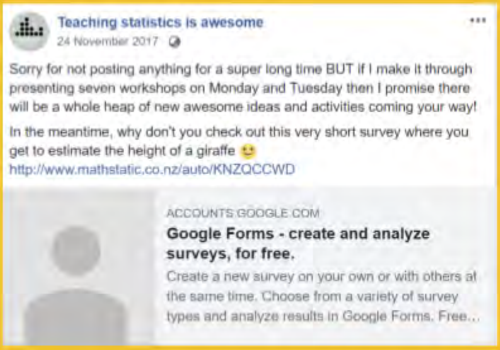
Random allocation



Code.gs

```
function doGet() {
var webpages = [];
//---------------just change the webpage URLs
//the webpages need to hosted on https sites - if you want more than two just create a new line using the same code as below
//just change the link to the google form
webpages.push("https://docs.google.com/forms/d/e/1FAIpQLSfLfuayN7YVhFRzAWG87ArEeML6ltEUgDOB5rA9F0KIH2u_5w/viewform?usp=sf_link");
webpages.push("https://docs.google.com/forms/d/e/1FAIpQLSdjjtjCVrovMpGDH-9J1BwmXI5kNgK_EmJM1sfBRFDw3C4wWg/viewform?usp=sf_link");
//----------------------------------------
var random_page = Math.floor(Math.random()*webpages.length);
var redirect = webpages[random_page];
return HtmlService.createHtmlOutput("<script>location.href = '" + redirect + "';</script>");
//line below requires authorisation but also works - thanks Nikolai Cook
//return HtmlService.createHtmlOutput(UrlFetchApp.fetch(redirect));
}
```

Help?

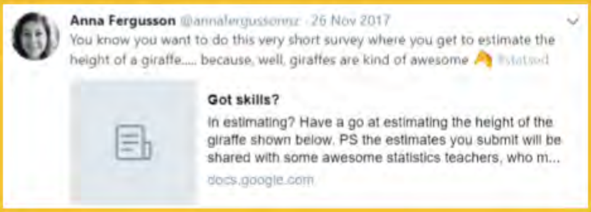


Teaching statistics is awesome
24 November 2017

Sorry for not posting anything for a super long time BUT if I make it through presenting seven workshops on Monday and Tuesday then I promise there will be a whole heap of new awesome ideas and activities coming your way!

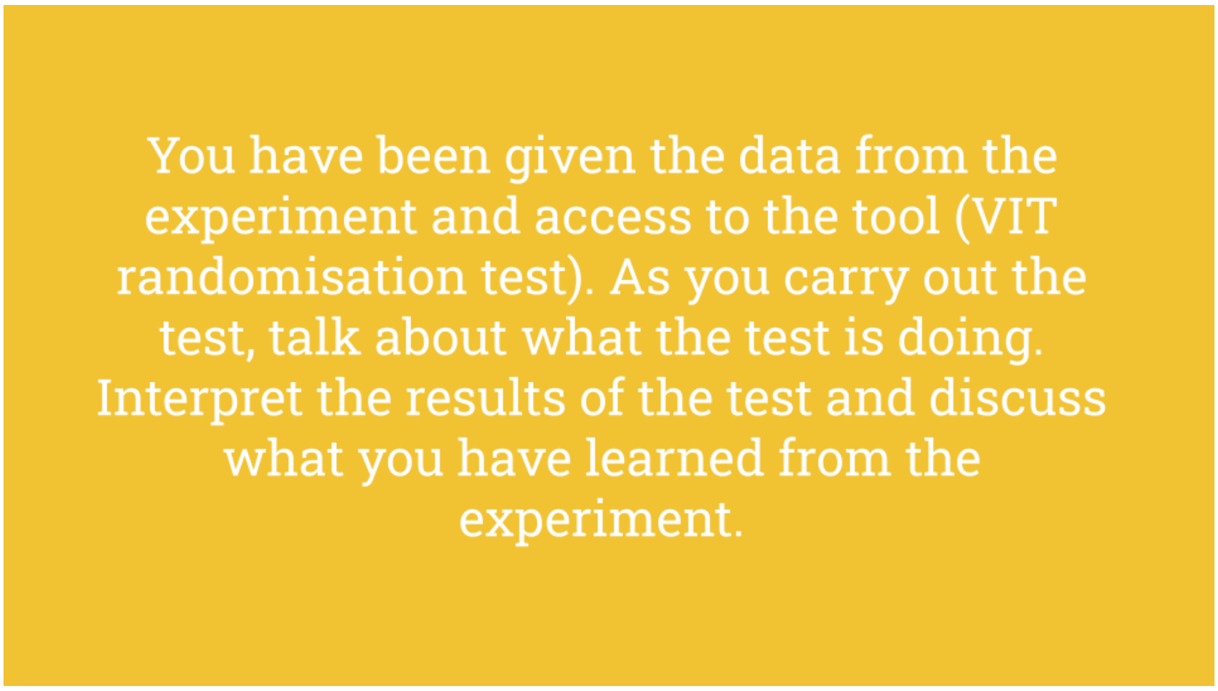
In the meantime, why don't you check out this very short survey where you get to estimate the height of a giraffe 😉
http://www.mathstatic.co.nz/auto/KNZQCCWD

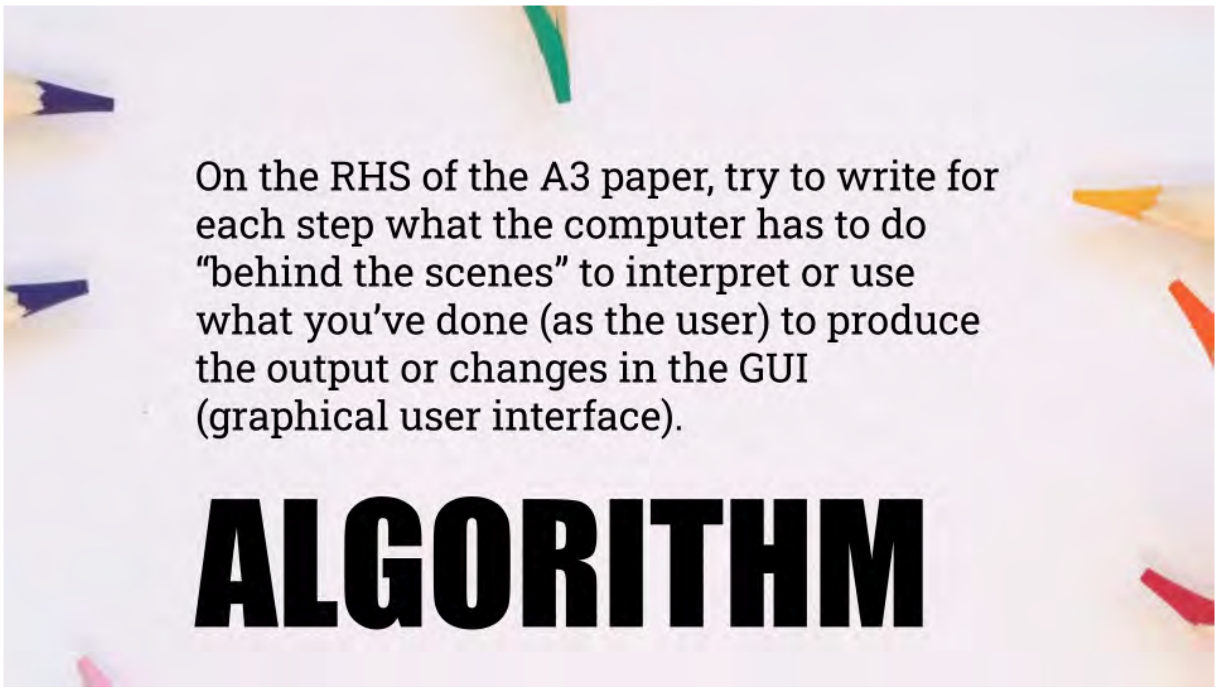
ACCOUNTS.GOOGLE.COM
Google Forms - create and analyze surveys, for free.
Create a new survey on your own or with others at the same time. Choose from a variety of survey types and analyze results in Google Forms. Free...



Anna Fergusson @annafergussonnz · 26 Nov 2017
You know you want to do this very short survey where you get to estimate the height of a giraffe..... because, well, giraffes are kind of awesome 🦒 #statsed

Got skills?
In estimating? Have a go at estimating the height of the giraffe shown below. PS the estimates you submit will be shared with some awesome statistics teachers, who m...
docs.google.com


You have been given the data from the experiment and access to the tool (VIT randomisation test). As you carry out the test, talk about what the test is doing. Interpret the results of the test and discuss what you have learned from the experiment.

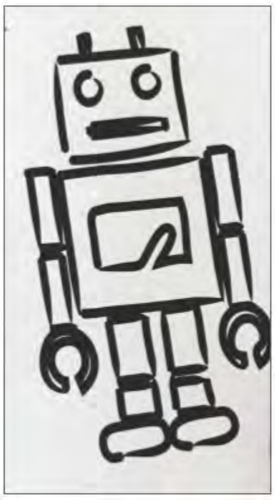

On the RHS of the A3 paper, try to write for each step what the computer has to do “behind the scenes” to interpret or use what you’ve done (as the user) to produce the output or changes in the GUI (graphical user interface).

ALGORITHM

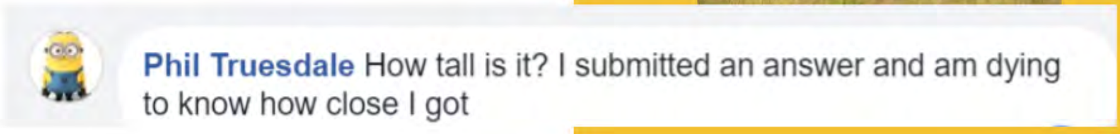
Let's learnr some more coding

Refresh the shiny app if needed.

Work through the task until you get to the "stop sign"!



How tall is this giraffe?

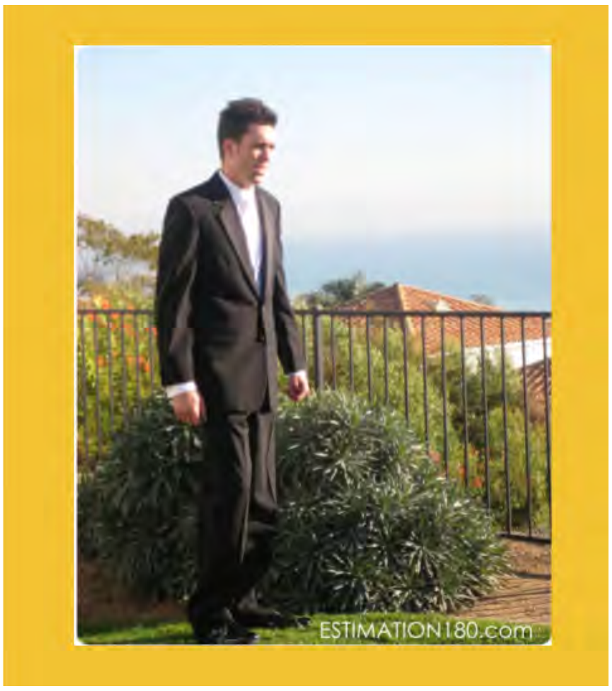

Phil Truesdale How tall is it? I submitted an answer and am dying to know how close I got

What is this guy's height?

What is a number that is too low?

What is a number that is too high?

What is your estimate?



Let's learnr some more coding

Back to the shiny app!

Work through the task until you get to the next "stop sign"!

Rmarkdown in RStudio

Time to get creative!

Go to RStudio Cloud (it's one of the tabs already set up). You might need to refresh the web browser.

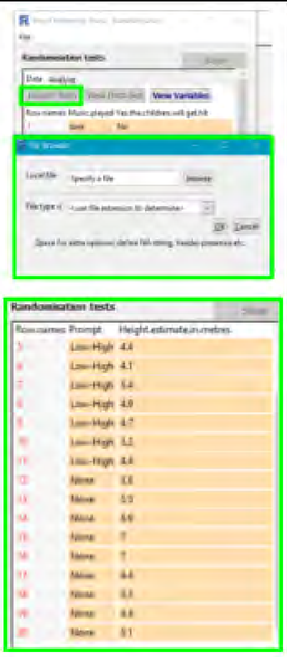
Follow the instructions on the RHS of the screen (the HTML output) to make changes to the LHS (the code).

Data sets

Giraffe data set: https://goo.gl/Dhx8Yy

Man data set: https://goo.gl/wMRbyx

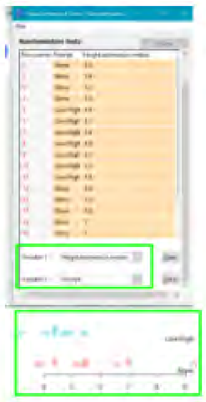
A3 sheet with screenshots from VIT Online



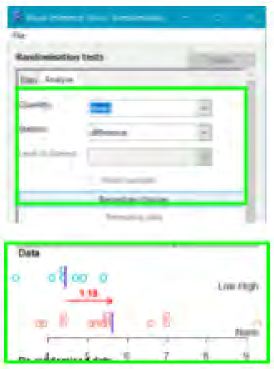
Randomisation tests

Row.names	Prompt	Height.estimate.in.metres
3	Low-High	4.4
4	Low-High	4.1
7	Low-High	5.4
8	Low-High	4.9
9	Low-High	4.7
10	Low-High	5.2
11	Low-High	4.4
12	None	3.8
13	None	5.5
14	None	6.9
15	None	7
16	None	7
17	None	6.4
18	None	6.3
19	None	6.8
20	None	5.1

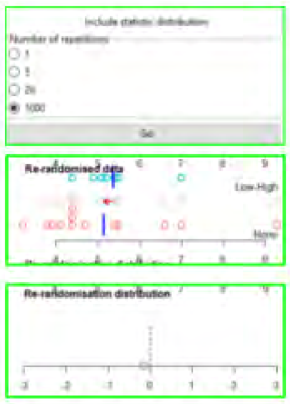
Step 1:



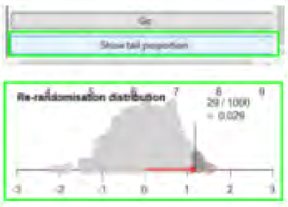
Step 2:



Step 3:



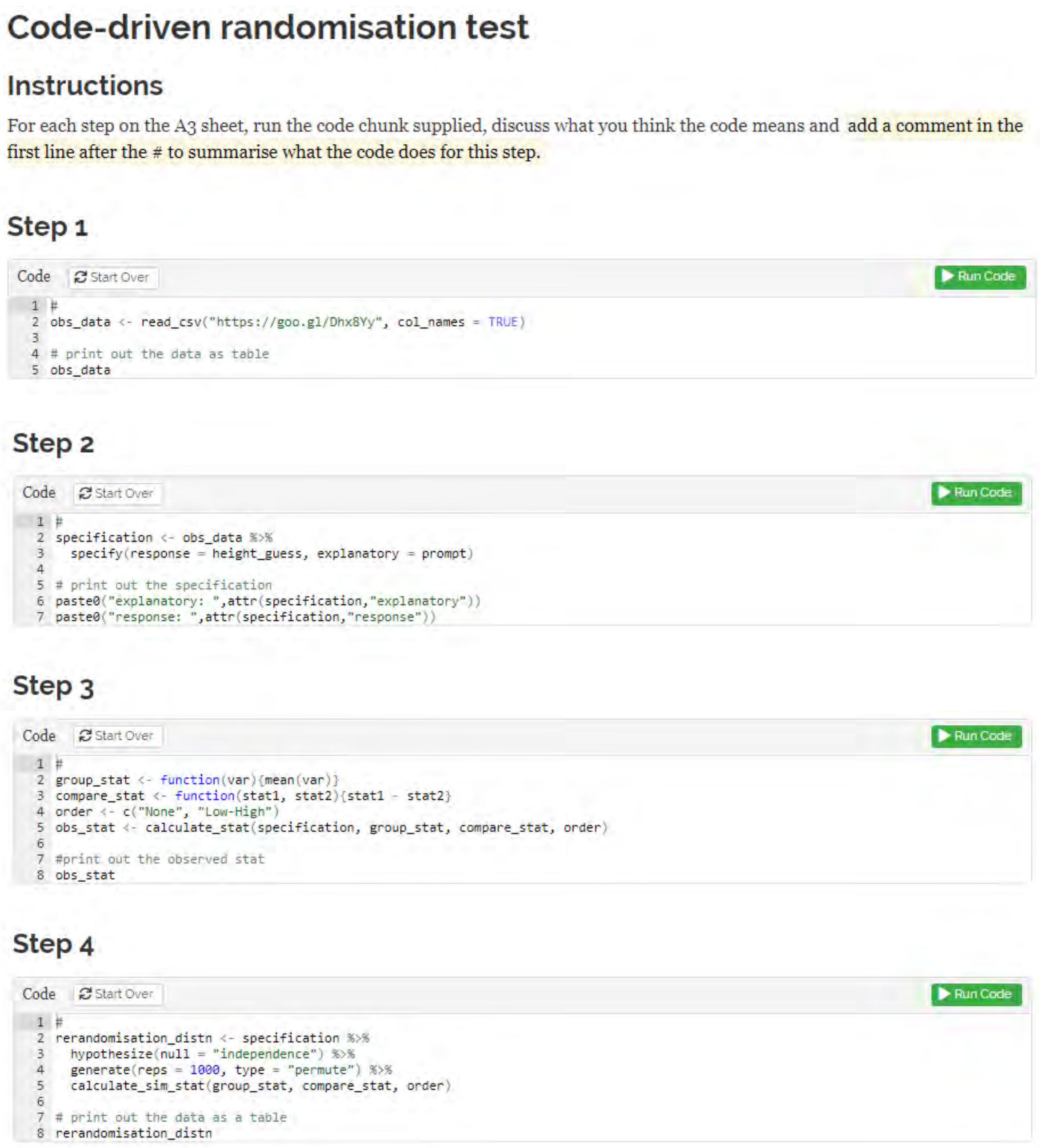
Step 4:



Step 5:

learnr activity

A static version of the web-based task is provided below.



Code-driven randomisation test

Instructions

For each step on the A3 sheet, run the code chunk supplied, discuss what you think the code means and add a comment in the first line after the # to summarise what the code does for this step.

Step 1

```r
#
obs_data <- read_csv("https://goo.gl/Dhx8Yy", col_names = TRUE)

# print out the data as table
obs_data
```

Step 2

```r
#
specification <- obs_data %>%
  specify(response = height_guess, explanatory = prompt)

# print out the specification
paste0("explanatory: ",attr(specification,"explanatory"))
paste0("response: ",attr(specification,"response"))
```

Step 3

```r
#
group_stat <- function(var){mean(var)}
compare_stat <- function(stat1, stat2){stat1 - stat2}
order <- c("None", "Low-High")
obs_stat <- calculate_stat(specification, group_stat, compare_stat, order)

#print out the observed stat
obs_stat
```

Step 4

```r
#
rerandomisation_distn <- specification %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate_sim_stat(group_stat, compare_stat, order)

# print out the data as a table
rerandomisation_distn
```

Step 5

```r
randomisation_distn %>%
  filter(sim_stat > obs_stat) %>%
  nrow()/1000
```

Wait for the next instructions from Anna

What about that guy?

The source of the data for the height guesses for the guy can be found here: https://goo.gl/wMRbyx

Try re-using the code from the steps above to carry out the randomisation test using code.

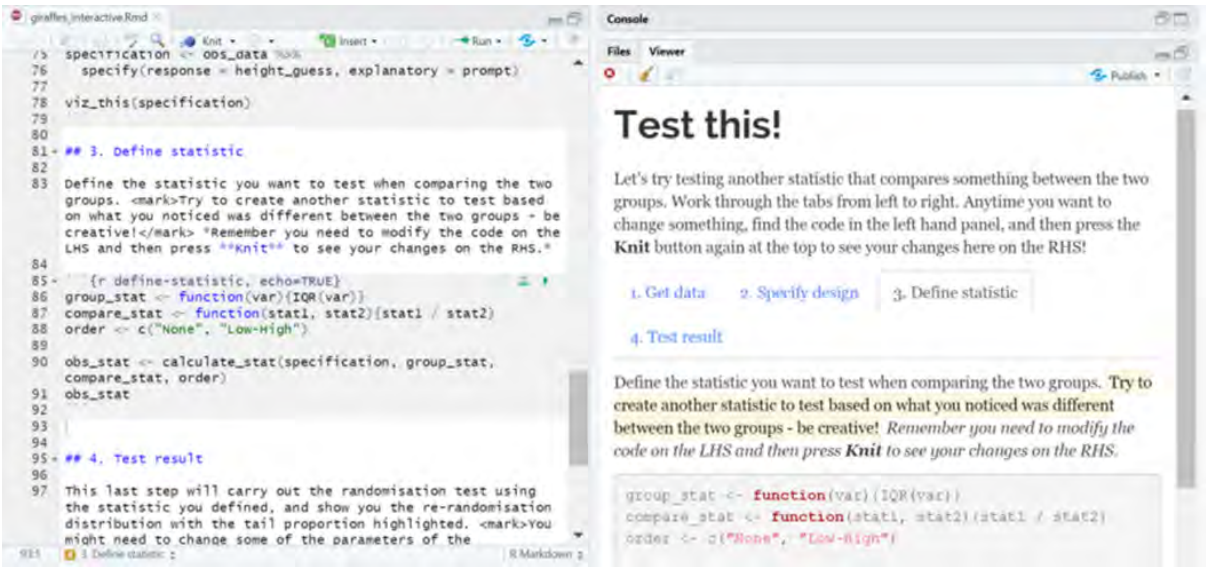
As you carry out the test, talk about what the test is doing.

Interpret the results of the test and discuss what you have learned from the experiment.

```r

```

RMarkdown activity

Screenshots from the RMarkdown document are provided below.



~~~
specification <- obs_data %>%
  specify(response = height_guess, explanatory = prompt)

viz_this(specification)


## 3. Define statistic

Define the statistic you want to test when comparing the two
groups. <mark>Try to create another statistic to test based
on what you noticed was different between the two groups - be
creative!</mark> *Remember you need to modify the code on the
LHS and then press **Knit** to see your changes on the RHS.*

```{r define-statistic, echo=TRUE}
group_stat <- function(var){IQR(var)}
compare_stat <- function(stat1, stat2){stat1 / stat2}
order <- c("None", "Low-High")

obs_stat <- calculate_stat(specification, group_stat,
compare_stat, order)
obs_stat



## 4. Test result

This last step will carry out the randomisation test using
the statistic you defined, and show you the re-randomisation
distribution with the tail proportion highlighted. <mark>You
might need to change some of the parameters of the
~~~

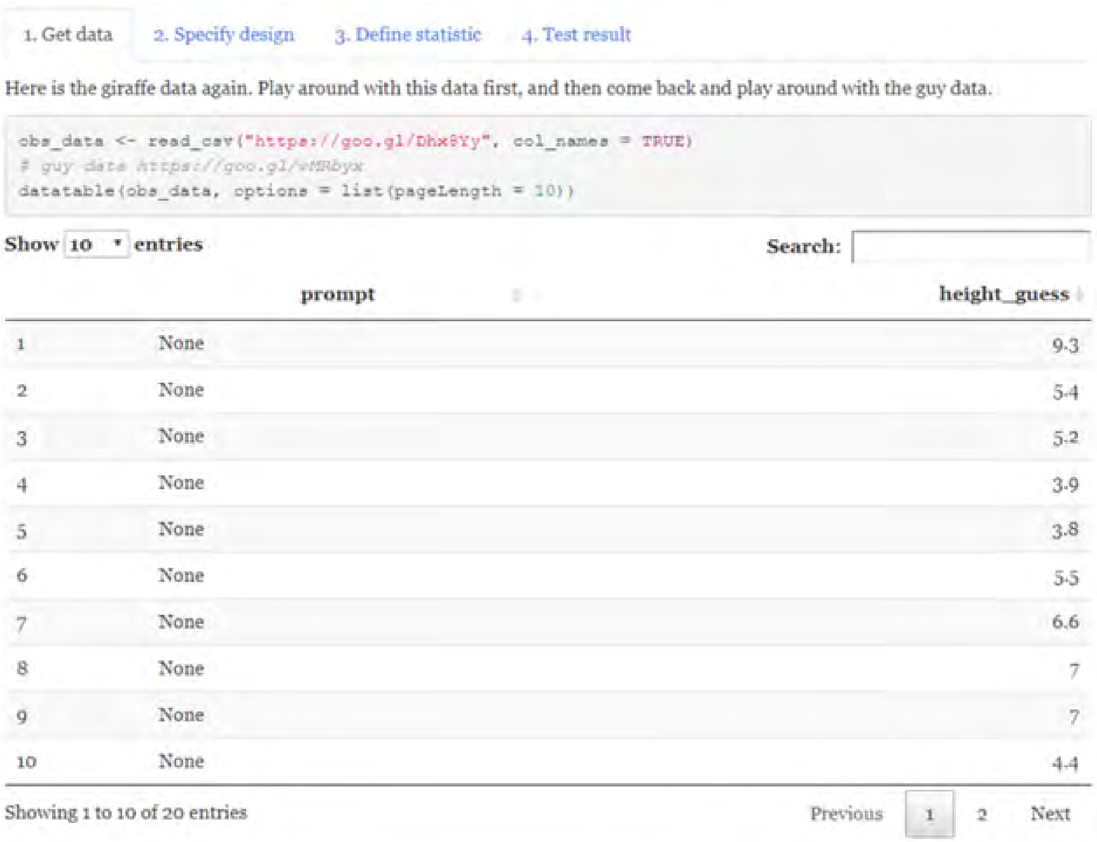
Test this!

Let's try testing another statistic that compares something between the two groups. Work through the tabs from left to right. Anytime you want to change something, find the code in the left hand panel, and then press the **Knit** button again at the top to see your changes here on the RHS!

1. Get data 2. Specify design 3. Define statistic 4. Test result

Define the statistic you want to test when comparing the two groups. Try to create another statistic to test based on what you noticed was different between the two groups - be creative! *Remember you need to modify the code on the LHS and then press* ***Knit*** *to see your changes on the RHS.*

```
group_stat <- function(var){IQR(var)}
compare_stat <- function(stat1, stat2){stat1 / stat2}
order <- c("None", "Low-High")
```



1. Get data 2. Specify design 3. Define statistic 4. Test result

Here is the giraffe data again. Play around with this data first, and then come back and play around with the guy data.

```
obs_data <- read_csv("https://goo.gl/Dhx8Yy", col_names = TRUE)
# guy data https://goo.gl/sMRbyx
datatable(obs_data, options = list(pageLength = 10))
```

Show 10 entries Search:

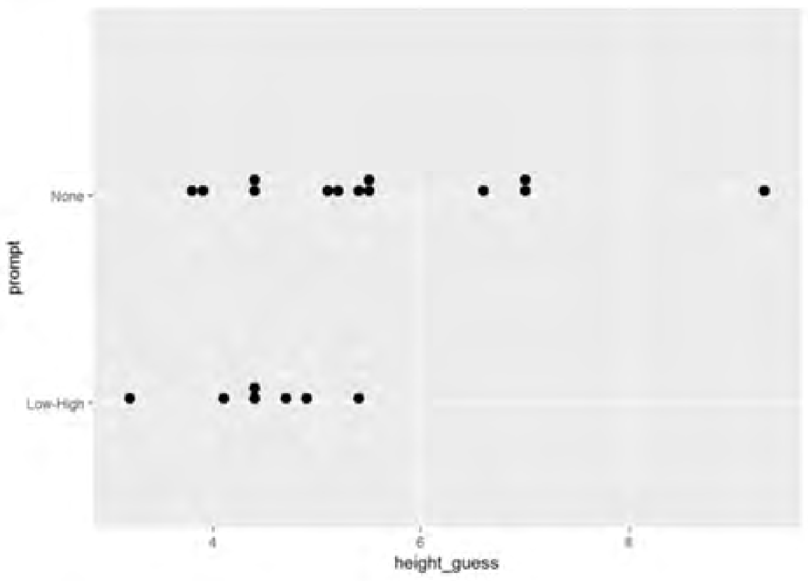
	prompt	height_guess
1	None	9.3
2	None	5.4
3	None	5.2
4	None	3.9
5	None	3.8
6	None	5.5
7	None	6.6
8	None	7
9	None	7
10	None	4.4

Showing 1 to 10 of 20 entries Previous 1 2 Next

1. Get data 2. Specify design 3. Define statistic 4. Test result

Decide which variables to use for the test. There's nothing here to change since only these two variables were used for the experiments! **But take another look at the data for each group and discuss how the height guesses for each group compare.**

```
specification <- obs_data %>%
  specify(response = height_guess, explanatory = prompt)

viz_this(specification)
```



1. Get data 2. Specify design 3. Define statistic 4. Test result

Define the statistic you want to test when comparing the two groups. **Try to create another statistic to test based on what you noticed was different between the two groups - be creative!** *Remember you need to modify the code on the LHS and then press **Knit** to see your changes on the RHS.*

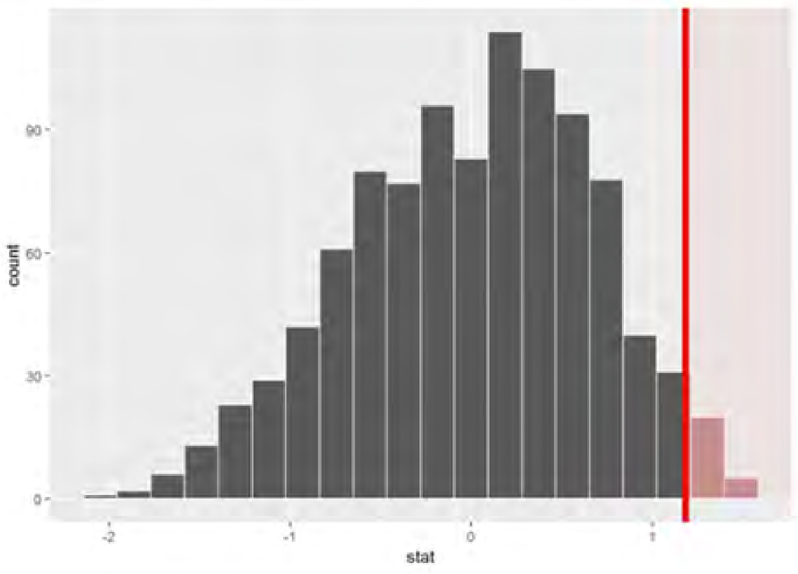
```
group_stat <- function(var){mean(var)}
compare_stat <- function(stat1, stat2){stat1 - stat2}
order <- c("None", "Low-High")

obs_stat <- calculate_stat(specification, group_stat, compare_stat, order)
obs_stat
```

```
## [1] 1.18022
```

1. Get data   2. Specify design   3. Define statistic   4. Test result

This last step will carry out the randomisation test using the statistic you defined, and show you the re-randomisation distribution with the tail proportion highlighted. **You might need to change some of the parameters of the visualise_distn function, depending on what you are testing.** *e.g. direction = "less" will shade the left tail, direction = "two_sided" will shade both tails, bins=n changes how many bins are used for the histogram*

```
rerandomisation_distn <- specification %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate_sim_stat(group_stat, compare_stat, order)

visualise_distn(rerandomisation_distn, bins=20, direction = "greater", obs_stat = obs_stat)
```

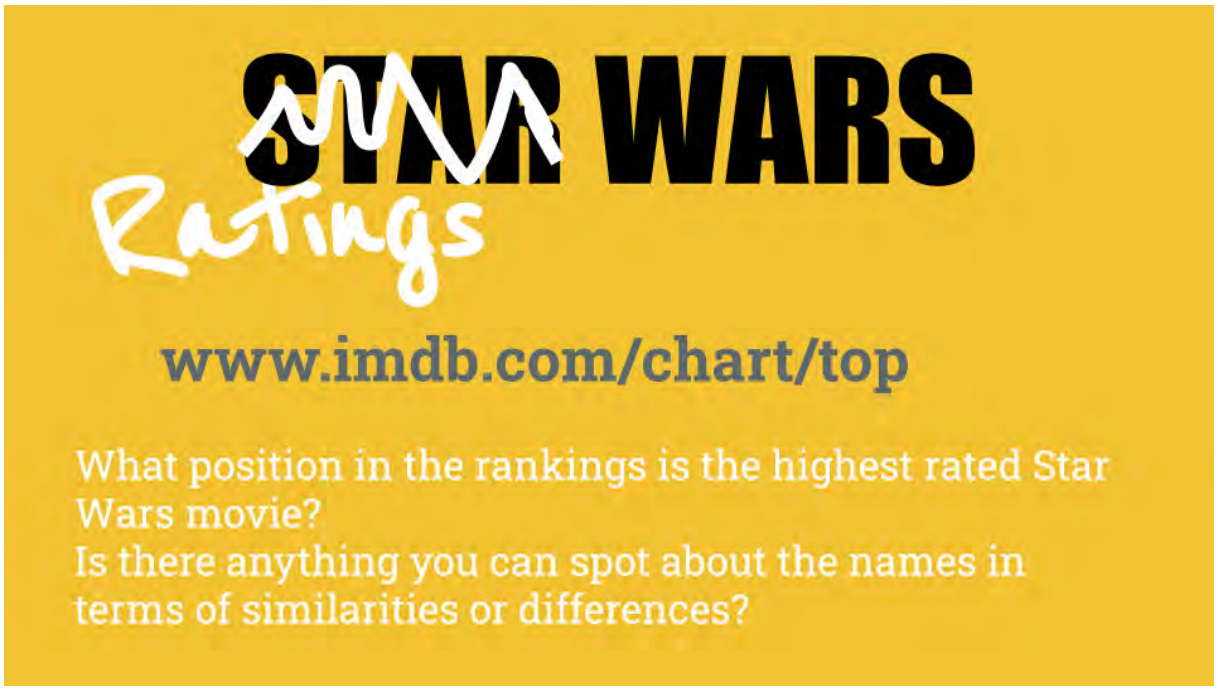


```
rerandomisation_distn %>%
  filter(sim_stat > obs_stat) %>%
  nrow()/1000
```

```
## [1] 0.025
```

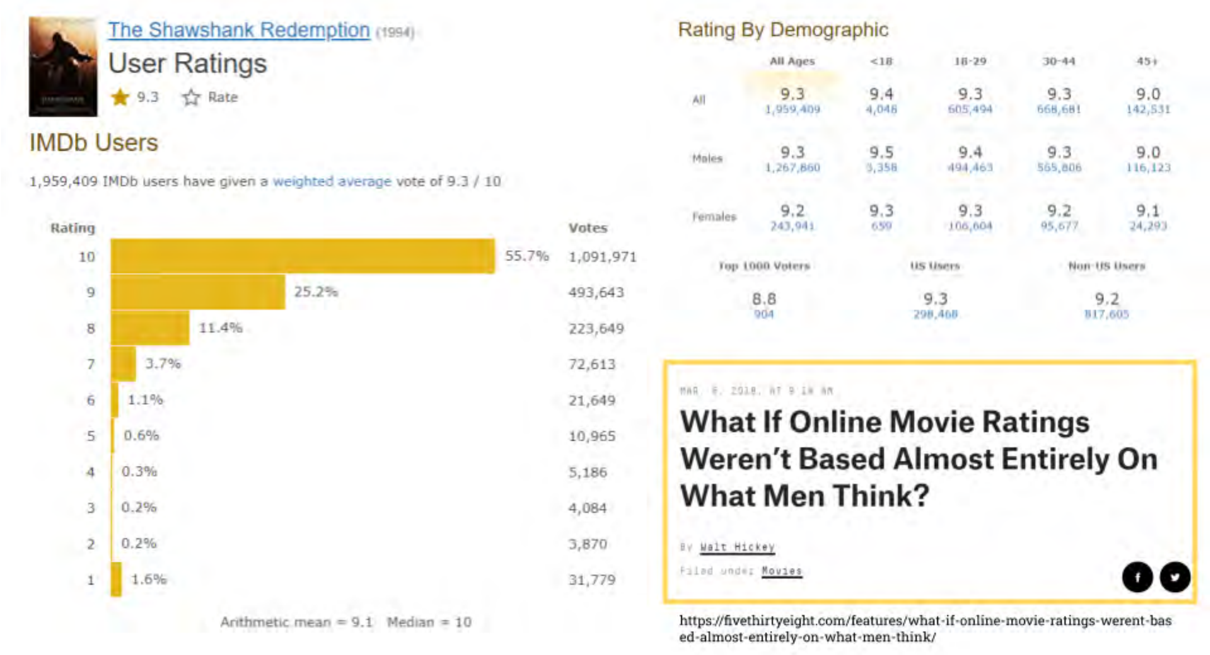
Appendix D - Prediction modelling task

Slides used at the start of the task


STAR WARS
Ratings

www.imdb.com/chart/top

What position in the rankings is the highest rated Star Wars movie?
Is there anything you can spot about the names in terms of similarities or differences?



The Shawshank Redemption (1994)
User Ratings
★ 9.3 ☆ Rate

IMDb Users

1,959,409 IMDb users have given a weighted average vote of 9.3 / 10

Rating		Votes
10	55.7%	1,091,971
9	25.2%	493,643
8	11.4%	223,649
7	3.7%	72,613
6	1.1%	21,649
5	0.6%	10,965
4	0.3%	5,186
3	0.2%	4,084
2	0.2%	3,870
1	1.6%	31,779

Arithmetic mean = 9.1 Median = 10

Rating By Demographic

	All Ages	<18	18-29	30-44	45+
All	9.3 1,959,409	9.4 4,048	9.3 605,494	9.3 668,681	9.0 142,531
Males	9.3 1,267,860	9.5 3,358	9.4 494,463	9.3 565,806	9.0 116,123
Females	9.2 243,941	9.3 659	9.3 106,604	9.2 95,677	9.1 24,293

Top 1000 Voters	US Users	Non-US Users
8.8 904	9.3 298,468	9.2 817,605

MAR. 6, 2018, AT 9:14 AM

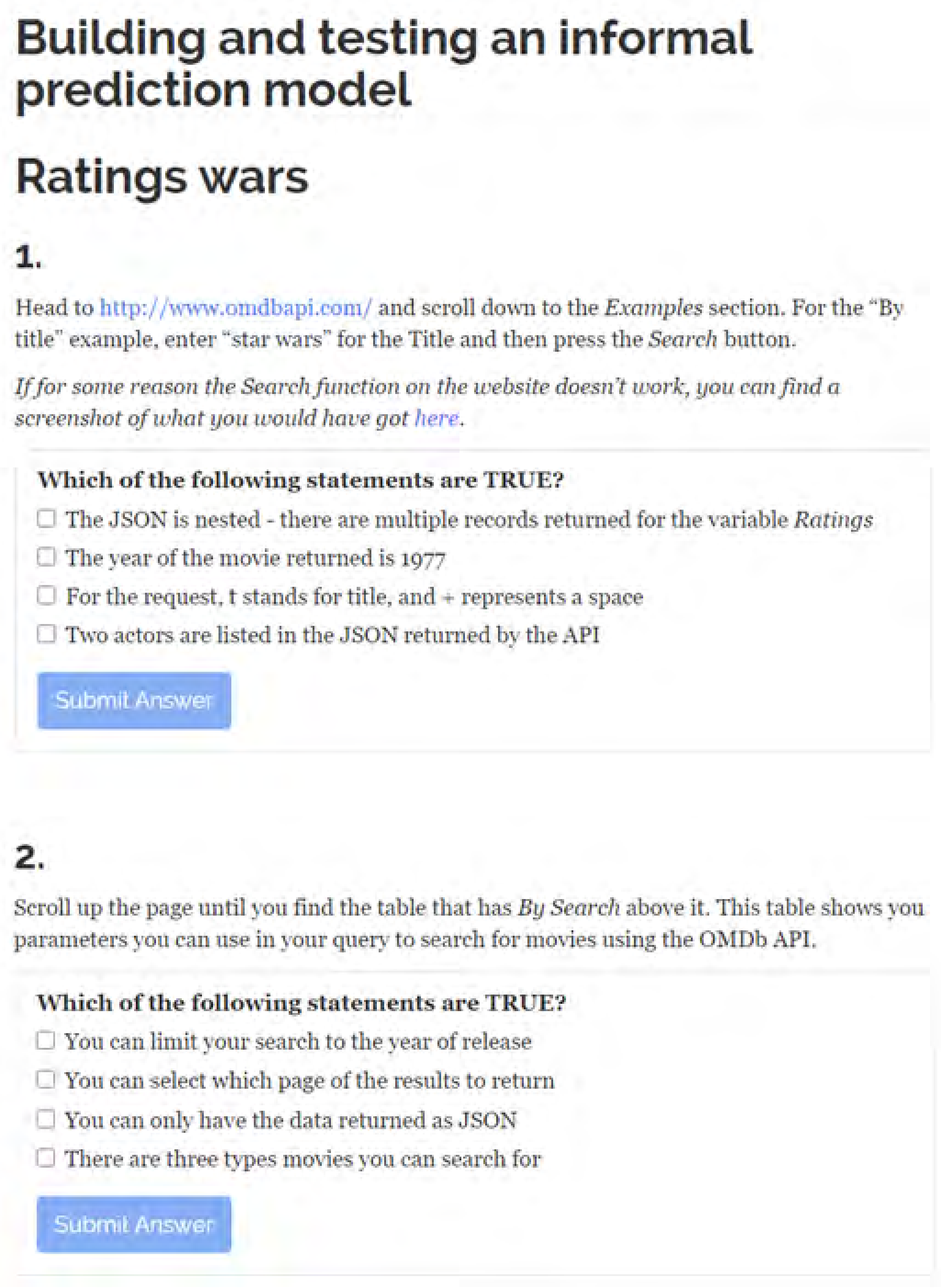
What If Online Movie Ratings Weren't Based Almost Entirely On What Men Think?

By Walt Hickey
Filed under Movies

https://fivethirtyeight.com/features/what-if-online-movie-ratings-werent-based-almost-entirely-on-what-men-think/

Web-based task

A static version of the web-based task is provided below.



Building and testing an informal prediction model

Ratings wars

1.

Head to http://www.omdbapi.com/ and scroll down to the *Examples* section. For the "By title" example, enter "star wars" for the Title and then press the *Search* button.

If for some reason the Search function on the website doesn't work, you can find a screenshot of what you would have got here.

Which of the following statements are TRUE?

- ☐ The JSON is nested - there are multiple records returned for the variable *Ratings*
- ☐ The year of the movie returned is 1977
- ☐ For the request, t stands for title, and + represents a space
- ☐ Two actors are listed in the JSON returned by the API

2.

Scroll up the page until you find the table that has *By Search* above it. This table shows you parameters you can use in your query to search for movies using the OMDb API.

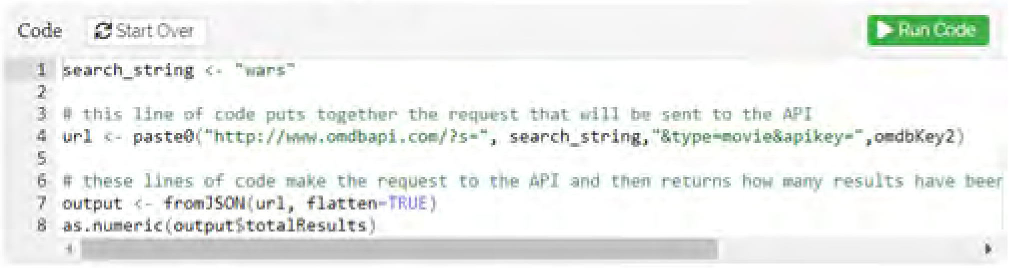
Which of the following statements are TRUE?

- ☐ You can limit your search to the year of release
- ☐ You can select which page of the results to return
- ☐ You can only have the data returned as JSON
- ☐ There are three types movies you can search for

3.

The code below makes a request to the OMDb API using some of the parameters from the *By search* table we just looked at. Try changing either the word(s) used in the search_string or parts of the url and running the code chunk, to answer the question that follows the code chunk.

Remember to use a + for a space



```r
search_string <- "wars"

# this line of code puts together the request that will be sent to the API
url <- paste0("http://www.omdbapi.com/?s=", search_string,"&type=movie&apikey=",omdbKey2)

# these lines of code make the request to the API and then returns how many results have beer
output <- fromJSON(url, flatten=TRUE)
as.numeric(output$totalResults)
```

Which of the following statements are TRUE?

☐ There are around 17 movies with both dog and cat in their title

☐ There are around 728 (TV) series with wars in their title

☐ There are around 2357 movies with star in their title

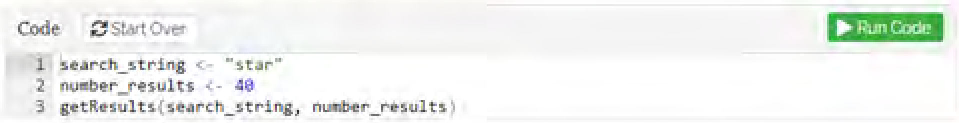
☐ There are more movies with cat in their title than dog

Submit Answer

4.

Let's get some data! The code below will request the first *n* results for whatever you search for, and display the results in a nice table.

Be patient - the code may take some time to run!



```r
search_string <- "star"
number_results <- 40
getResults(search_string, number_results)
```

Now have a go at this question, which may require some code tinkering!

Which of the following statements are TRUE?

☐ The last variable/column is called **DVD**

☐ You can't request any more than 100 results with this function

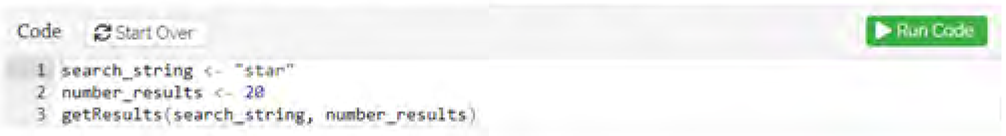
☐ There are 10 rows (or movies) on each 'page' of the data table

☐ The 68th movie when you search for star is called Lego Star Wars: The Padawan Menace

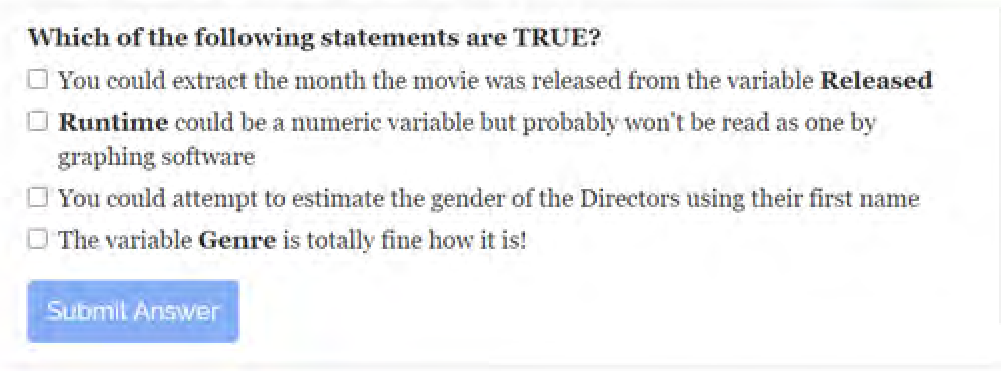
Submit Answer

5.

In today's task, we won't focus too much on data manipulation or cleaning. But take another look at some of those variables to answer the question below!



```r
search_string <- "star"
number_results <- 20
getResults(search_string, number_results)
```



Which of the following statements are TRUE?

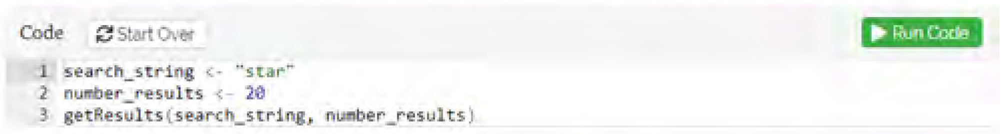
- ☐ You could extract the month the movie was released from the variable **Released**
- ☐ **Runtime** could be a numeric variable but probably won't be read as one by graphing software
- ☐ You could attempt to estimate the gender of the Directors using their first name
- ☐ The variable **Genre** is totally fine how it is!

6.

Our goal for this task is to build an informal prediction model that uses the **imdbRating** for a movie to predict the **metascoreRating** for a movie.

Read some more about how IMDb calculates their ratings for movies and how it is different from how metacritic calculates their ratings (called a metascore).

Thinking about our morning session on data structures, take yet another look at the data below and discuss with each other how each variable has been created or measured.



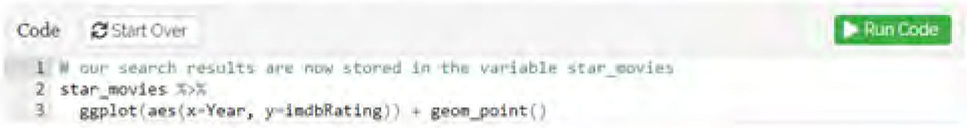
```r
search_string <- "star"
number_results <- 20
getResults(search_string, number_results)
```

Discuss with each other whether you expect there will be relationship between these two types of ratings for movies.

7.

Let's see what we can learn from the data! The code below will display a scatterplot, but not of the two variables we want! Run the code and discuss what you see .

Change the code so that it shows **metascoreRating** on the y-axis (the response variable) and **imdbRating** on the x-axis (the explanatory variable).

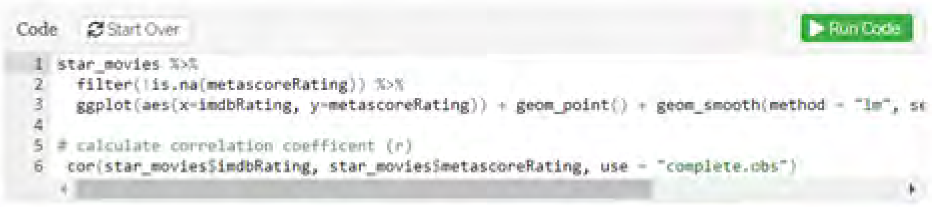


```r
# our search results are now stored in the variable star_movies
star_movies %>%
  ggplot(aes(x=Year, y=imdbRating)) + geom_point()
```

Discuss with each other what the plot reveals about the relationship between the **metascoreRating** and the **imdbRating** for these movies.

8.

Run the code below which will fit a line to the data and output the correlation coefficient for the two variables.



```r
star_movies %>%
  filter(!is.na(metascoreRating)) %>%
  ggplot(aes(x=imdbRating, y=metascoreRating)) + geom_point() + geom_smooth(method = "lm", se

# calculate correlation coefficent (r)
 cor(star_movies$imdbRating, star_movies$metascoreRating, use = "complete.obs")
```

Discuss with each other how well you think the line models the relationship between the **metascoreRating** and the **imdbRating** for these movies.

9.

Suppose we wanted to predict the metascoreRating of another movie that had a imdbRating of 7.

Using the data and line displayed in the plot in the previous section, answer the question below.

Which of the following statements are TRUE?

☐ Movies with the same imdbRatings can have different metascoreRatings

☐ You can use an interval to give the predicted metascore rating based on the variation (vertical scatter) observed in the data/plot

☐ You would be just as confident making predictions using imdbRatings of 3 or lower as you would making predictions using imdbRatings of between 5 to 9

☐ Using the line, you could predict the metascoreRating to be around 62

Submit Answer

Discuss with each other what two numbers you could use for a prediction interval of the **metascoreRating** of another movie that had a **imdbRating** of 7.

10.

We are now going to build an informal prediction model that has the form:

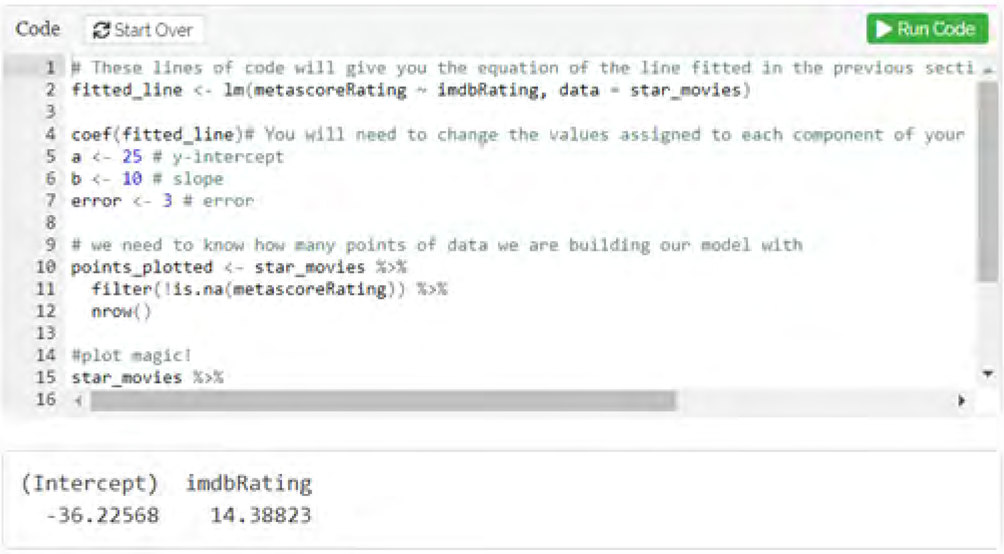
> predicted **metascoreRating** = a + b * **imdbRating** ± error

a is the y-intercept and *b* is the slope of the "line of best fit" (a simple linear regression model).

error will be based on your visual estimate of how far away the points sit vertically from the line.

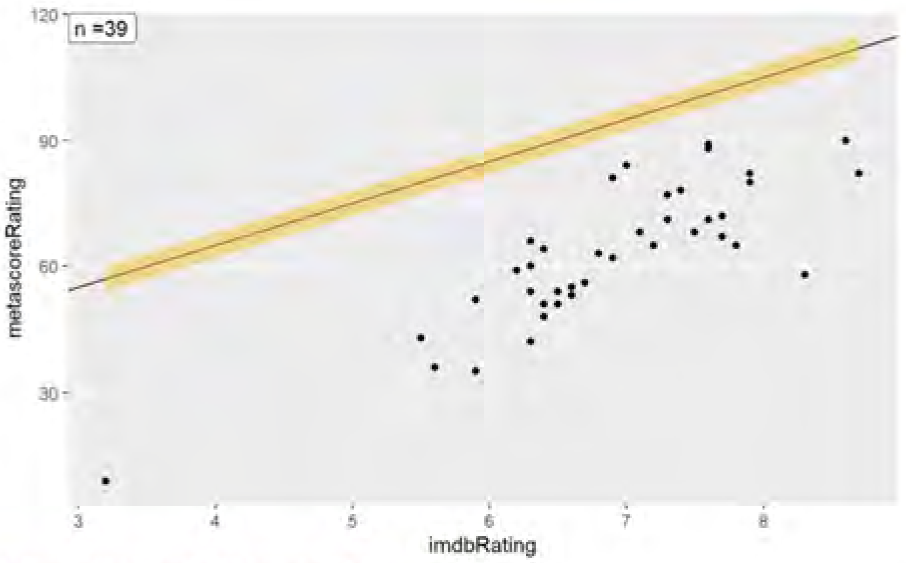
This model will give you a prediction interval for the **metascoreRating** using a movie's **imdbRating**.

Run the code below and then read the comments within the code to learn how to build your informal prediction model. Adjust your model until you are happy with it.



```
# These lines of code will give you the equation of the line fitted in the previous secti
fitted_line <- lm(metascoreRating ~ imdbRating, data = star_movies)

coef(fitted_line)# You will need to change the values assigned to each component of your
a <- 25 # y-intercept
b <- 10 # slope
error <- 3 # error

# we need to know how many points of data we are building our model with
points_plotted <- star_movies %>%
  filter(!is.na(metascoreRating)) %>%
  nrow()

#plot magic!
star_movies %>%
```

```
(Intercept)  imdbRating
  -36.22568    14.38823
```



Discuss the following with each other:

- If you only use the fitted line to make predictions, what percentage of movies have their **metascoreRating** correctly predicted from their **imdbRating**?
- If you use your informal prediction model to make predictions instead, what percentage of movies have their **metascoreRating** correctly predicted from their **imdbRating**?

11.

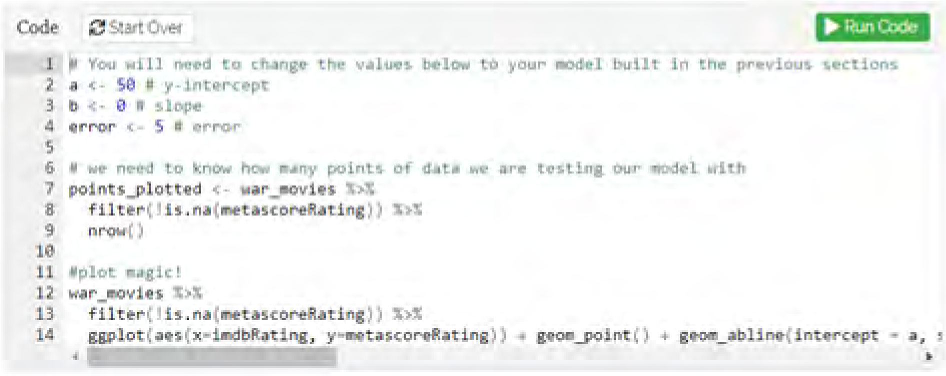
Time to test that model of yours out!

Before we do, discuss with each other how well you think your model will work to predict the **metascoreRating** from a movie's **imdbRating** for movies with war in their title.

PS - you've still got time to change your model!

12.

In the code below, add the details of your model BEFORE running the code (no sneaky peeks!)



```
# You will need to change the values below to your model built in the previous sections
a <- 50 # y-intercept
b <- 0 # slope
error <- 5 # error

# we need to know how many points of data we are testing our model with
points_plotted <- war_movies %>%
  filter(!is.na(metascoreRating)) %>%
  nrow()

#plot magic!
war_movies %>%
  filter(!is.na(metascoreRating)) %>%
  ggplot(aes(x=imdbRating, y=metascoreRating)) + geom_point() + geom_abline(intercept = a,
```

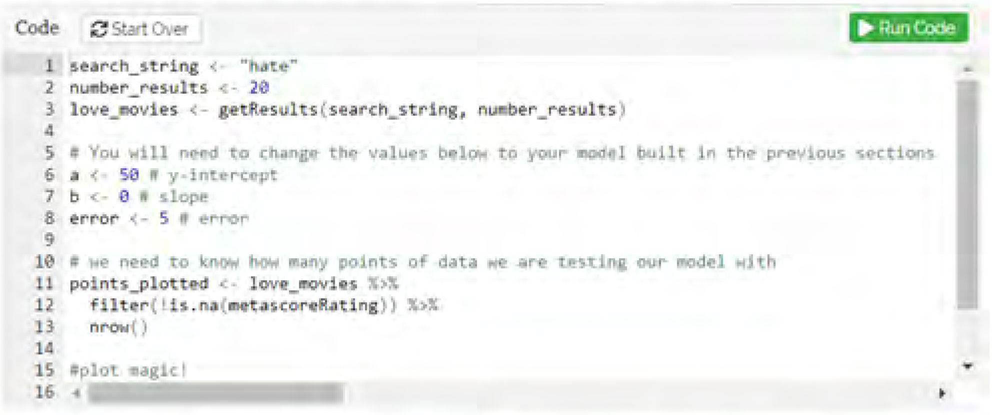
Discuss with each other how well your model performed on the test data.

13.

The final battle of the ratings!

Love is better than war, so let's try out your model on **50** movies with the love in their title.

This time you are going to have to change a few things in the code before you can test your model. Remember you can scroll back up to previous sections to see what was done there!



```
search_string <- "hate"
number_results <- 20
love_movies <- getResults(search_string, number_results)

# You will need to change the values below to your model built in the previous sections
a <- 50 # y-intercept
b <- 0 # slope
error <- 5 # error

# we need to know how many points of data we are testing our model with
points_plotted <- love_movies %>%
  filter(!is.na(metascoreRating)) %>%
  nrow()

#plot magic!
```

Discuss with each other how well your model performed on this test data. You can also discuss any potential issues you noticed during the modelling process and what you might do differently next time to help deal with these issues.

14.

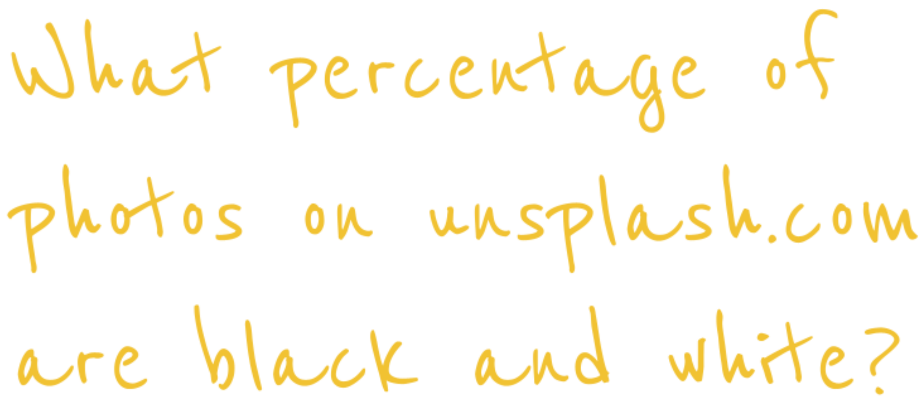
And we're done!

If there's still time, why don't you scroll back up the page and take a closer look at the code used for each section? Talk with your partner and see if you can work out what is going on.

Feel free to tinker with the code - you can't break anything, and ctrl+z will undo any changes you make :-)

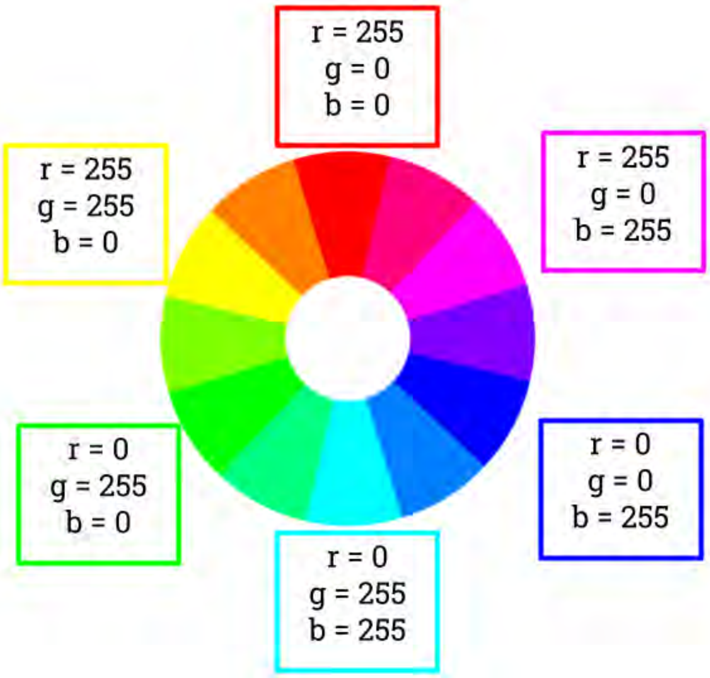
Appendix E - Classification modelling task

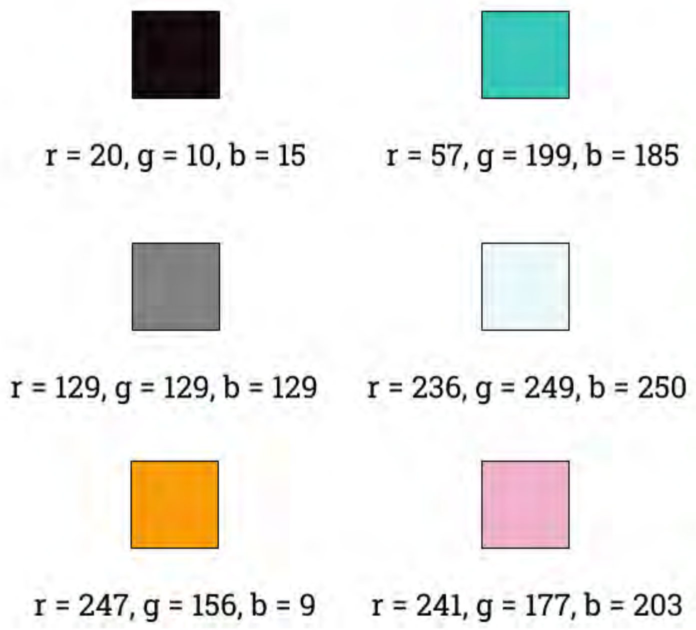
Slides used during task



What percentage of photos on unsplash.com are black and white?

https://en.wikipedia.org/wiki/Jelly_bean



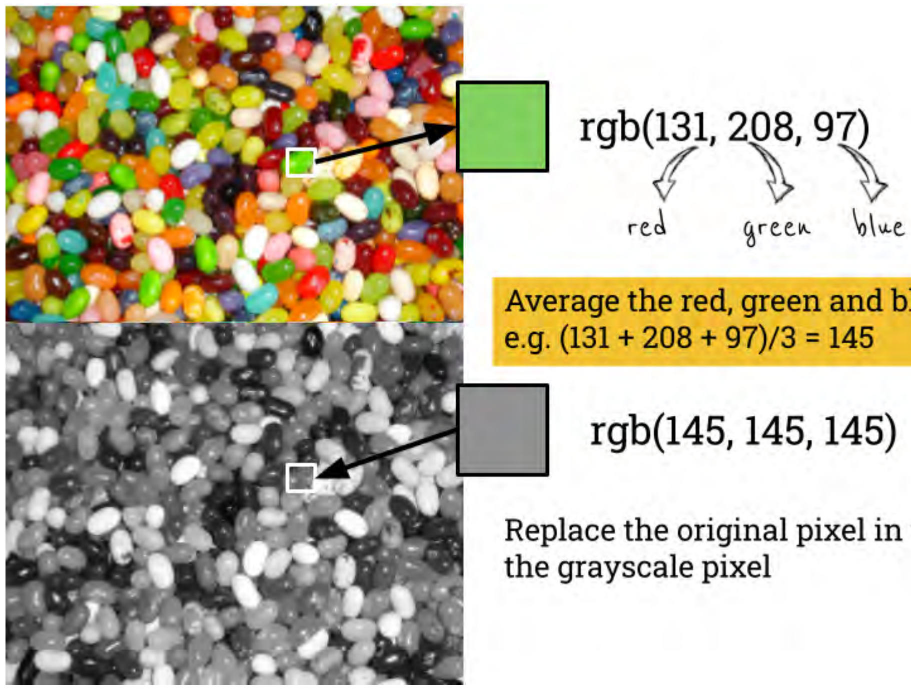

r = 20, g = 10, b = 15
r = 57, g = 199, b = 185
r = 129, g = 129, b = 129
r = 236, g = 249, b = 250
r = 247, g = 156, b = 9
r = 241, g = 177, b = 203

How do you get dark colours?

How do you get light colours?

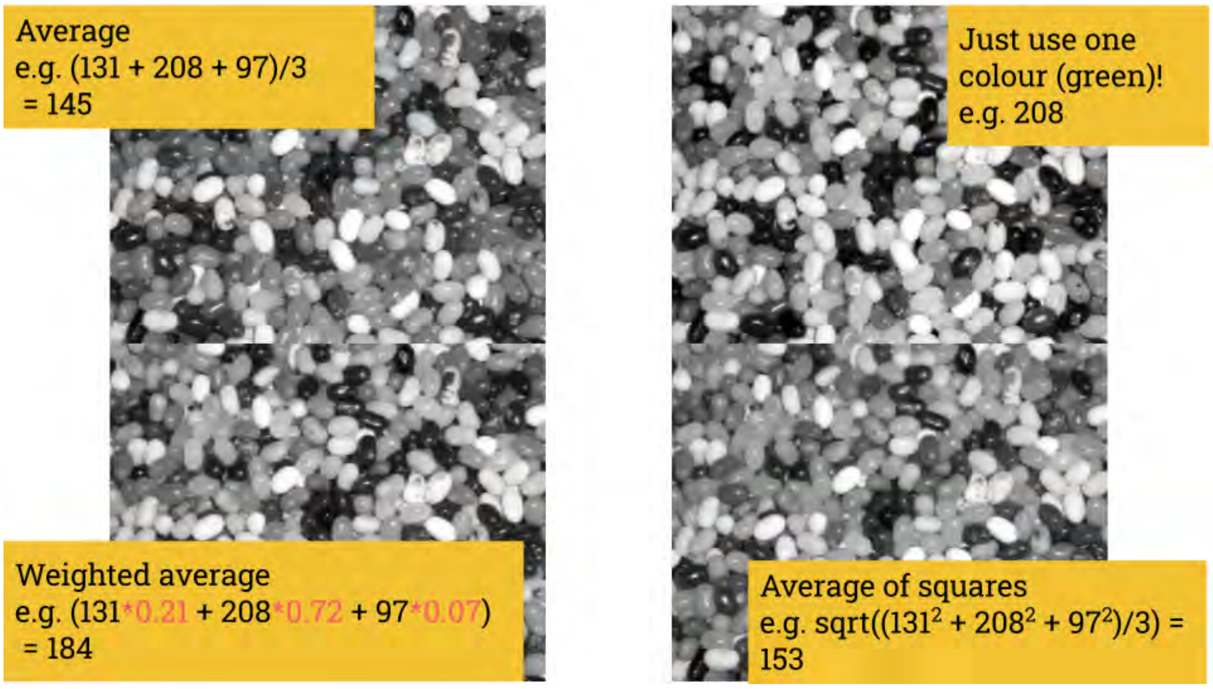
Does having a red component make something look red?

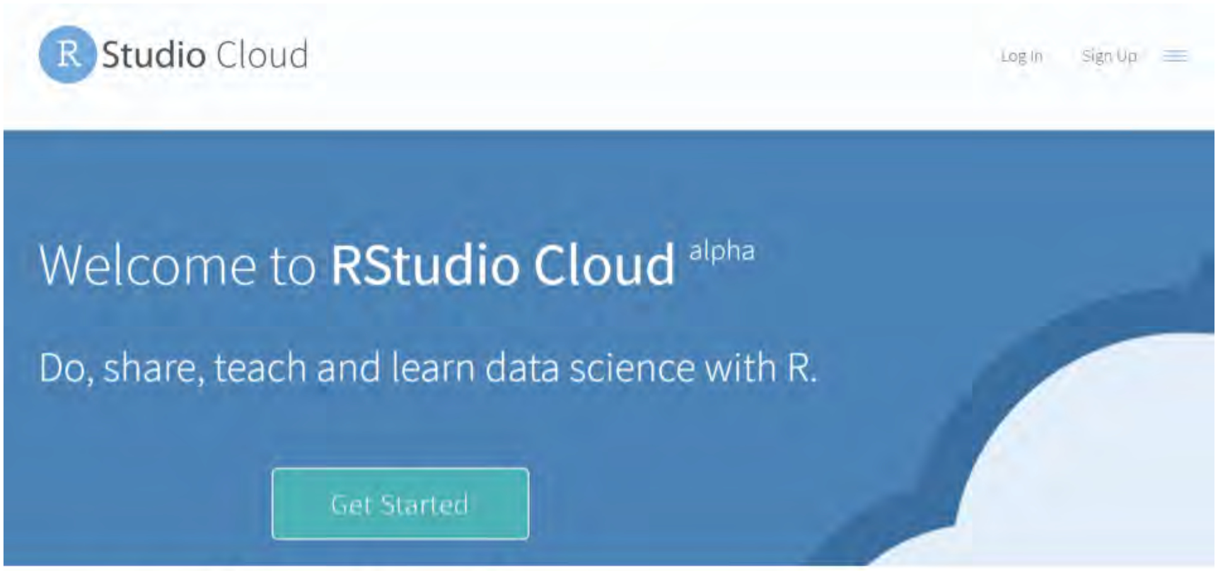
How do you get a shade of grey?



Average the red, green and blue components
e.g. (131 + 208 + 97)/3 = 145

Replace the original pixel in the photo with the grayscale pixel





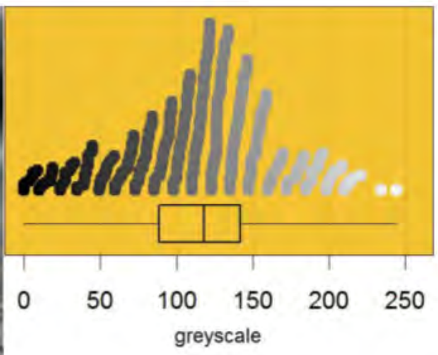
Welcome to RStudio Cloud alpha

Do, share, teach and learn data science with R.

rstudio.cloud

photo + plot

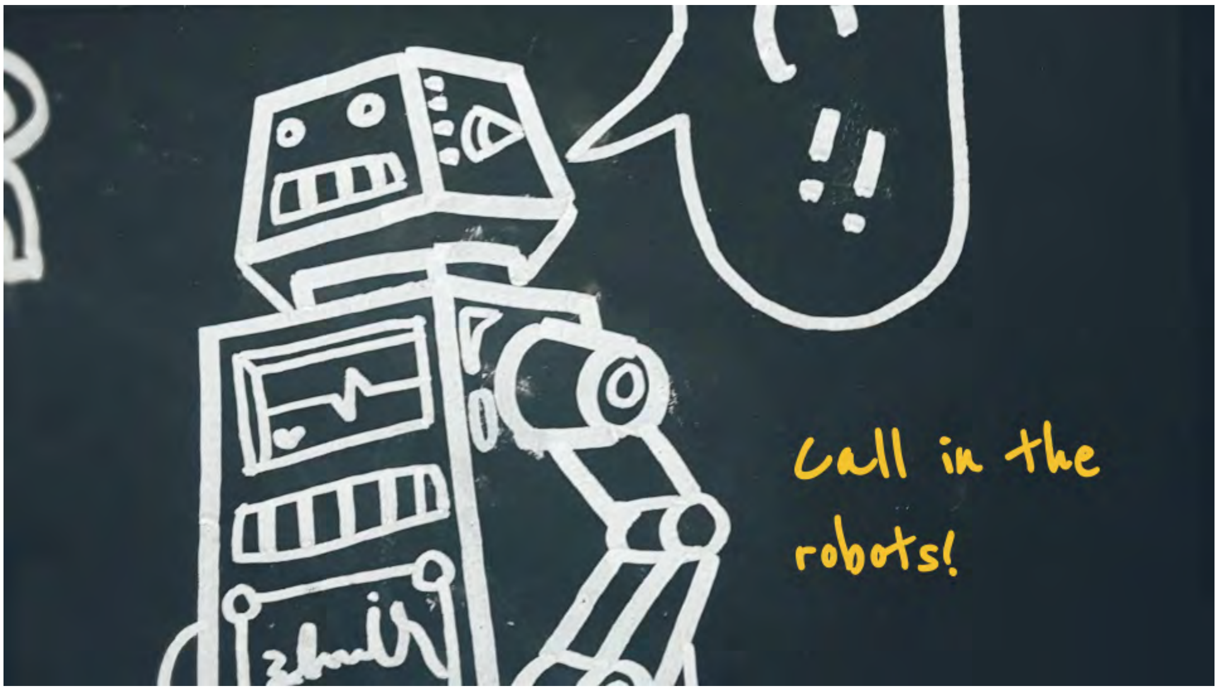
Try to match up the 10 black and white photos with their grayscale plots - discuss how you are making these matches



Most of these photos are not even close to normal!

light vs dark

Based on the your visual assessment of the each PHOTO, arrange the photo/plots from darkest to lightest



classification model

Develop a model to sort the photos into light or dark based on the value of the median greyscale

Good model?

Test it out by swapping your photo/plots with the pair to the left of you

What if we used the mean?

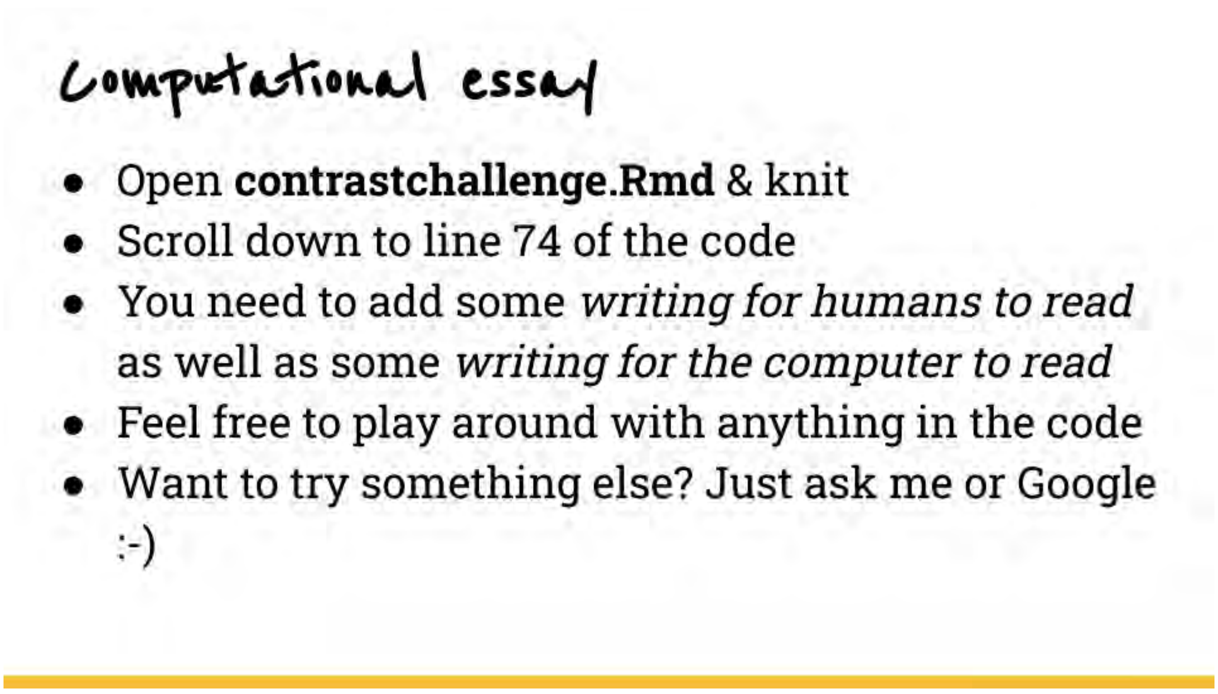
Discuss the advantages and/or disadvantages of using the mean for your model

RStudio Cloud - meanvsmedian.Rmd

- To run a "code chunk" press the green arrow at the top right of the chunk
- Fixups:
  - We want to take random sample of **500** pixels
  - We want to use the mean
  - We want to use your model
- Try it out your model with different numbered photos

classification model take two

Develop a model to sort the photos into high or low contrast

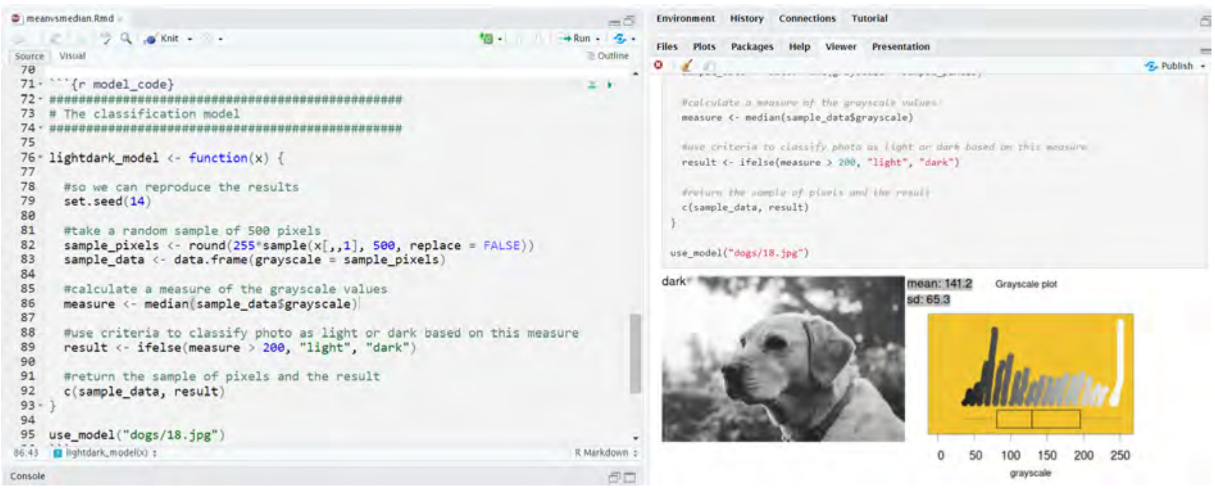


Computational essay

- Open **contrastchallenge.Rmd** & knit
- Scroll down to line 74 of the code
- You need to add some *writing for humans to read* as well as some *writing for the computer to read*
- Feel free to play around with anything in the code
- Want to try something else? Just ask me or Google :-)

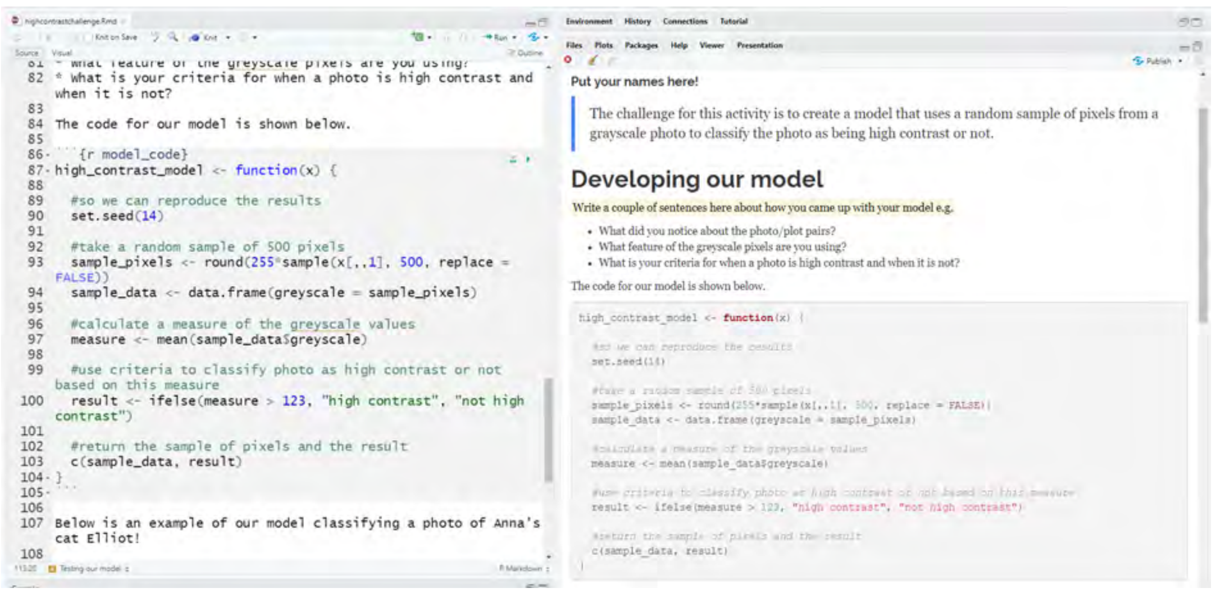
RMarkdown activity

Screenshot from the RMarkdown document used to explore using the mean or median to classify photos as light or dark (`meanvsmedian.Rmd`)



~~~
```{r model_code}
##################################################
# The classification model
##################################################

lightdark_model <- function(x) {

  #so we can reproduce the results
  set.seed(14)

  #take a random sample of 500 pixels
  sample_pixels <- round(255*sample(x[,,1], 500, replace = FALSE))
  sample_data <- data.frame(grayscale = sample_pixels)

  #calculate a measure of the grayscale values
  measure <- median(sample_data$grayscale)

  #use criteria to classify photo as light or dark based on this measure
  result <- ifelse(measure > 200, "light", "dark")

  #return the sample of pixels and the result
  c(sample_data, result)
}

use_model("dogs/18.jpg")
~~~

Screenshot from the RMarkdown document used to develop a classification model for high contrast photos (`contrastchallenge.Rmd`).



~~~
* What feature of the greyscale pixels are you using?
* What is your criteria for when a photo is high contrast and when it is not?

The code for our model is shown below.

```{r model_code}
high_contrast_model <- function(x) {

  #so we can reproduce the results
  set.seed(14)

  #take a random sample of 500 pixels
  sample_pixels <- round(255*sample(x[,,1], 500, replace = FALSE))
  sample_data <- data.frame(greyscale = sample_pixels)

  #calculate a measure of the greyscale values
  measure <- mean(sample_data$greyscale)

  #use criteria to classify photo as high contrast or not based on this measure
  result <- ifelse(measure > 123, "high contrast", "not high contrast")

  #return the sample of pixels and the result
  c(sample_data, result)
}
```

Below is an example of our model classifying a photo of Anna's cat Elliot!
~~~

Put your names here!

> The challenge for this activity is to create a model that uses a random sample of pixels from a grayscale photo to classify the photo as being high contrast or not.

Developing our model

Write a couple of sentences here about how you came up with your model e.g.

- What did you notice about the photo/plot pairs?
- What feature of the greyscale pixels are you using?
- What is your criteria for when a photo is high contrast and when it is not?

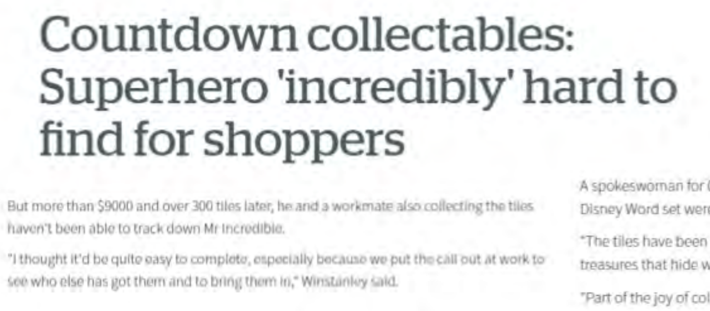
The code for our model is shown below.

Appendix F - Probability modelling task

Slides used during task

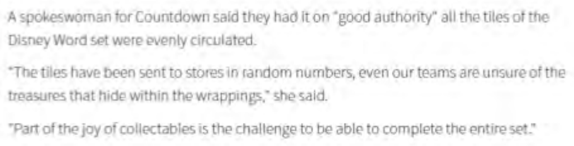
What are your experiences with collecting?

Discuss with your partner: Have you ever collected cards/tiles/etc. as part of competition or promotion?


Countdown collectables: Superhero 'incredibly' hard to find for shoppers

But more than $9000 and over 300 tiles later, he and a workmate also collecting the tiles haven't been able to track down Mr Incredible.

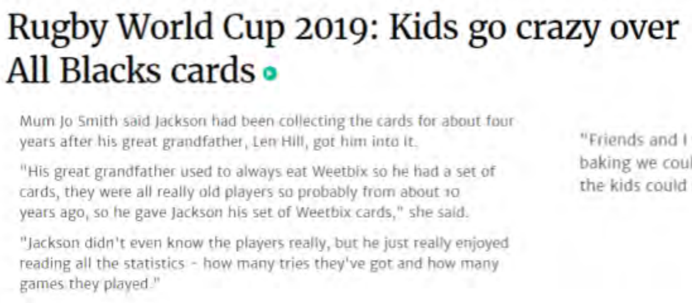
"I thought it'd be quite easy to complete, especially because we put the call out at work to see who else has got them and to bring them in," Winstanley said.


A spokeswoman for Countdown said they had it on "good authority" all the tiles of the Disney Word set were evenly circulated.

"The tiles have been sent to stores in random numbers, even our teams are unsure of the treasures that hide within the wrappings," she said.

"Part of the joy of collectables is the challenge to be able to complete the entire set."

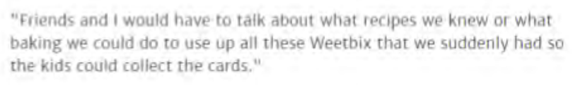
https://www.nzherald.co.nz/nz/news/article.cfm?c_id=1&objectid=12277768


Rugby World Cup 2019: Kids go crazy over All Blacks cards

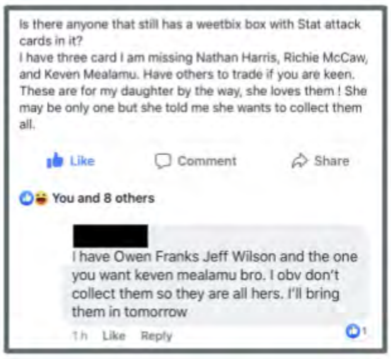
Mum Jo Smith said Jackson had been collecting the cards for about four years after his great grandfather, Len Hill, got him into it.

"His great grandfather used to always eat Weetbix so he had a set of cards, they were all really old players so probably from about 10 years ago, so he gave Jackson his set of Weetbix cards," she said.

"Jackson didn't even know the players really, but he just really enjoyed reading all the statistics - how many tries they've got and how many games they played."

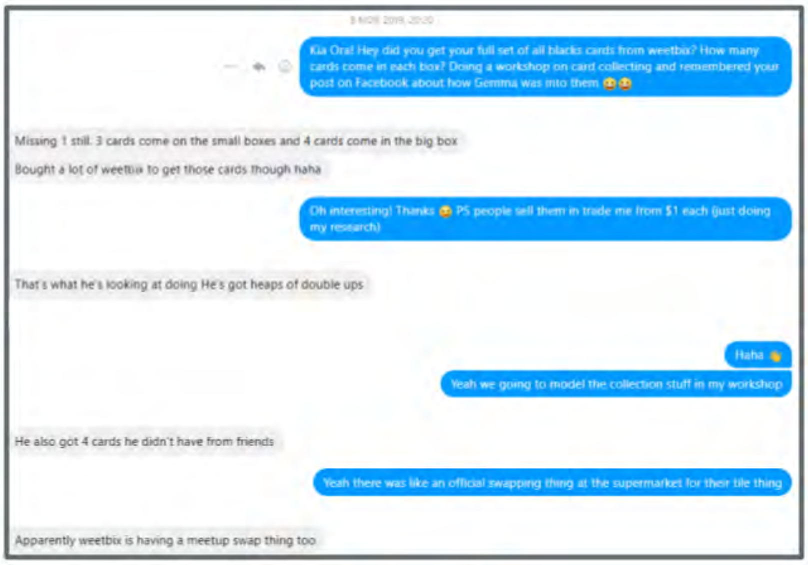

"Friends and I would have to talk about what recipes we knew or what baking we could do to use up all these Weetbix that we suddenly had so the kids could collect the cards."

https://www.stuff.co.nz/life-style/food-wine/116140952/rugby-world-cup-2019-kids-go-crazy-over-all-blacks-cards



Is there anyone that still has a weetbix box with Stat attack cards in it?
I have three card I am missing Nathan Harris, Richie McCaw, and Keven Mealamu. Have others to trade if you are keen. These are for my daughter by the way, she loves them ! She may be only one but she told me she wants to collect them all.

[redacted]
I have Owen Franks Jeff Wilson and the one you want keven mealamu bro. I obv don't collect them so they are all hers. I'll bring them in tomorrow



3 NOV 2019, 20:20

Kia Ora! Hey did you get your full set of all blacks cards from weetbix? How many cards come in each box? Doing a workshop on card collecting and remembered your post on Facebook about how Gemma was into them 😩😩

Missing 1 still. 3 cards come on the small boxes and 4 cards come in the big box

Bought a lot of weetbix to get those cards though haha

Oh interesting! Thanks 😩 PS people sell them in trade me from $1 each (just doing my research)

That's what he's looking at doing He's got heaps of double ups

Haha 👏

Yeah we going to model the collection stuff in my workshop

He also got 4 cards he didn't have from friends

Yeah there was like an official swapping thing at the supermarket for their tile thing

Apparently weetbix is having a meetup swap thing too

How about those fruity freezes?

Read the NZQA exemplar task "Fruity Freezes" and (re)familiarise yourself with the requirements of the task.

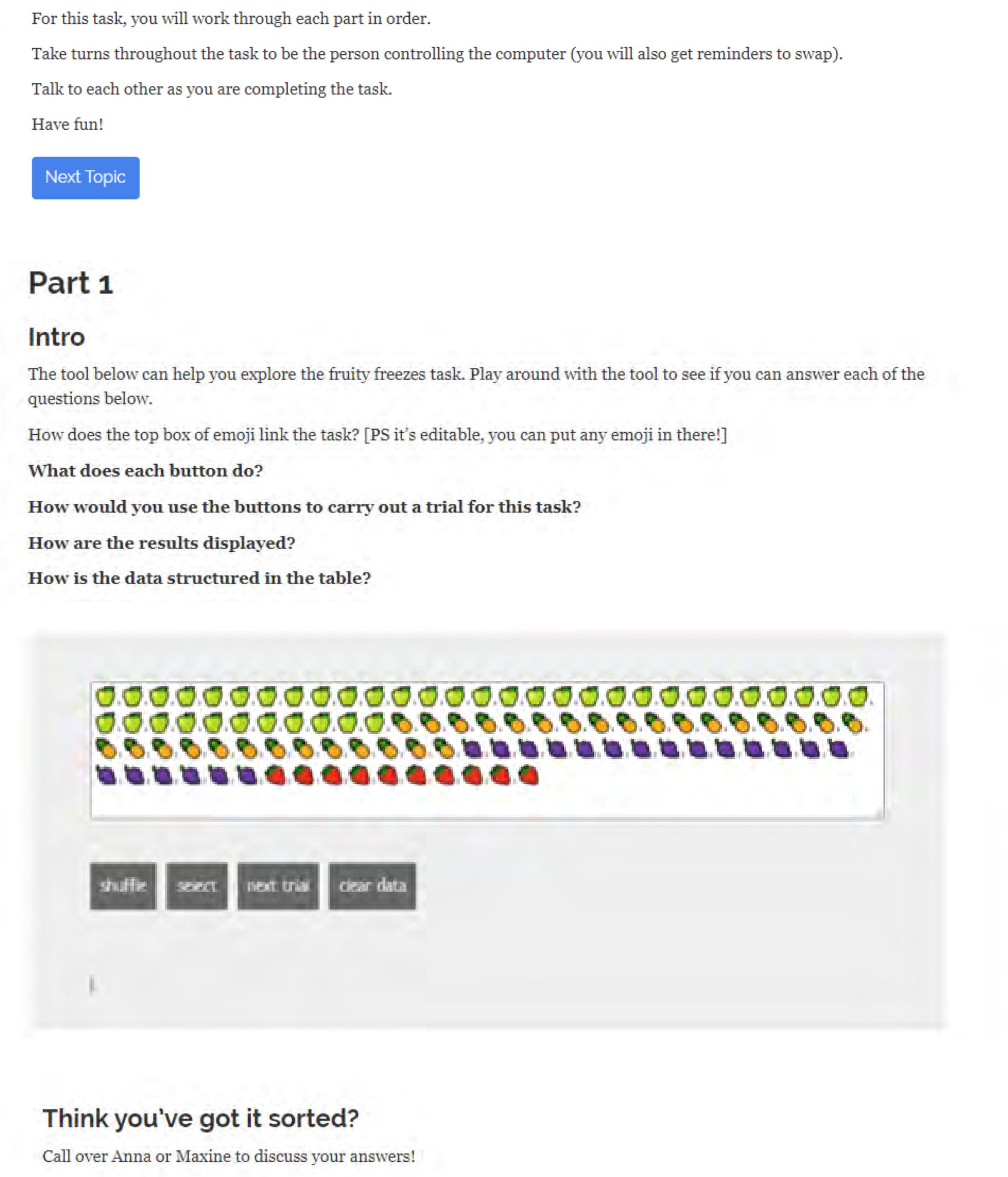
Discuss with your partner how you would teach students to complete a task like this.

Over the rest of today, you will work through the task independently in your groups.

Depending on what happens, I might bring us all back together at different points in the task :-)

Web-based task

A static version of the task is shown below.



For this task, you will work through each part in order.

Take turns throughout the task to be the person controlling the computer (you will also get reminders to swap).

Talk to each other as you are completing the task.

Have fun!

Part 1

Intro

The tool below can help you explore the fruity freezes task. Play around with the tool to see if you can answer each of the questions below.

How does the top box of emoji link the task? [PS it's editable, you can put any emoji in there!]

What does each button do?

How would you use the buttons to carry out a trial for this task?

How are the results displayed?

How is the data structured in the table?

Think you've got it sorted?

Call over Anna or Maxine to discuss your answers!

Part 2

If you haven't already, swap control of the computer :-)

Intro

We're now going to use a software tool called CODAP to model the situation from the fruity freezes task.

One of the many cool things about CODAP is that you can set up links with models, data and/or plots ready to use!

Getting to know CODAP

Below is a link to CODAP, with the simulation partially set up.

Open the link and press the start button.

https://codap.concord.org/releases/latest/static/dg/en/cert/index.html#shared=134988

Is the situation being modelled as you want for this task?

Take a look at the "Options" tab of the sampler tool and see if you can find a way to better model the situation!

Looking at the data

Below is a link to CODAP, with the simulation of 100 trials completed.

Open the link and take a look at the data generated.

https://codap.concord.org/releases/latest/static/dg/en/cert/index.html#shared=1349910

Click on the first row of the data table on the left.

What do the columns `num_values` , `unique_values` , `num_unique` and `prize` represent?

Note: Unfortunately, CODAP does not yet have a function that will count the number of unique values, so the formula I have used is hideous and is just a temporary workaround!

Analysing the results

To make a plot in CODAP, click the graph button on the top left menu, and then click an axis or drag one of the variables (attributes) from the data table onto the axis.

Drag `num_values` to the x-axis

To display the mean, click anywhere on the graph, and click the ruler button on the right hand side. Choose the mean option, and you will see a vertical blue line appear. Hover your mouse over this blue line and the mean will be displayed.

Try to create another plot, but this time one that displays the proportion of trials for which a prize was won!

Think you've got it sorted?

Although CODAP is not the focus for this workshop, feel free to play around with CODAP for a few minutes.

You could try making other plots from the data (HINT: Try making one with `unique_values`) or you could try putting your own emoji in the sampler tool :-)

Note: This version of the sampler tool has been created for this research workshop and I'll provide a link after the workshop so you can use this version.

Previous Topic Next Topic

Part 3

Intro

Ask Anna or Maxine for a set of screenshots taken while using CODAP for the fruity freezes task.

Arrange the screenshots in order of how you would use CODAP to investigate the fruity freezes task.

In the space beside each screenshot/step, write brief notes about what is happening statistically and/or computationally, and how this links to the fruity freezes task.

Think you've got it sorted?

Call over Anna or Maxine to discuss what you did!

Previous Topic Next Topic

Part 4

Swap control of the computer for this next part :-)

Intro

We are now going to use the programming language R to model the situation in the fruity freezes task.

You will be given a chunk of code for each step captured in the screenshots.

The code chunks will reveal how to carry out a simulation and analyse the data for the "Fruity Freezes" task using code.

For each code chunk that is revealed, match it to one of the cards you have beside you and add short comments in the code that describes what you think the code does (use # at the start of the line where you want to write a comment).

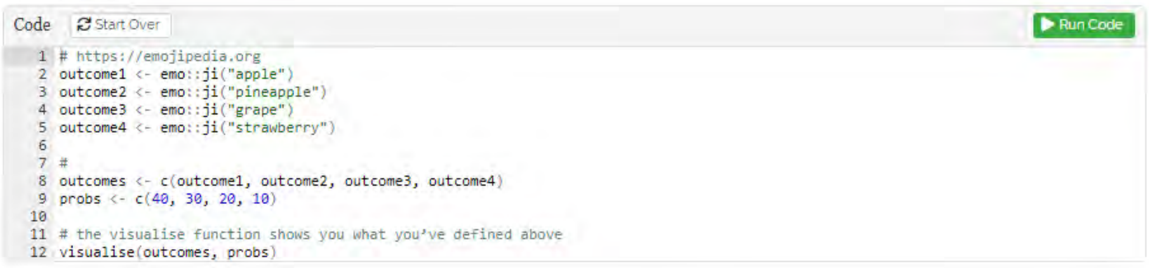
You might find it useful to run the code to see what the code produces before writing your comment(s)!

But first....

STOP TRYING TO MAKE MISTAKES HAPPEN
THEY ARE NEVER GOING TO HAPPEN
makeameme.org

Well, actually you'll probably make mistakes with coding. You might type something and the computer can't understand what you've written. But it's no big deal! You can press the "Start Over" button and try again :-)

Step 1

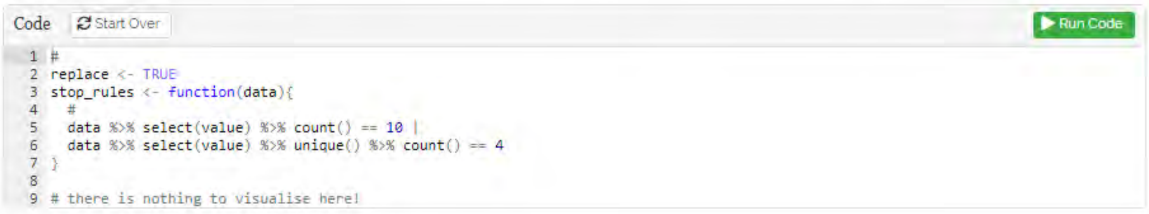


```
# https://emojipedia.org
outcome1 <- emo::ji("apple")
outcome2 <- emo::ji("pineapple")
outcome3 <- emo::ji("grape")
outcome4 <- emo::ji("strawberry")

#
outcomes <- c(outcome1, outcome2, outcome3, outcome4)
probs <- c(40, 30, 20, 10)

# the visualise function shows you what you've defined above
visualise(outcomes, probs)
```

Step 2

Note: In the code below, a `|` is how you do an "OR" and `%>%` is called a pipe!

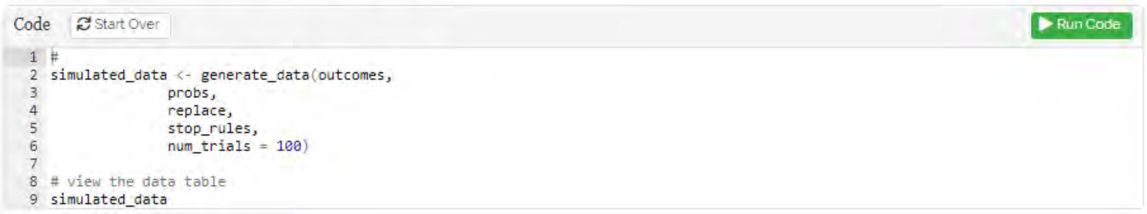
Since you are maths teachers, a pipe is like a composite function, except rather than nesting functions, you present these in the order you evaluate them e.g. `f(g(x))` *becomes* `x %>% g() %>% f()`. *With students, I just say the pipe stands for 'and then'*



```
#
replace <- TRUE
stop_rules <- function(data){
  #
  data %>% select(value) %>% count() == 10 |
  data %>% select(value) %>% unique() %>% count() == 4
}

# there is nothing to visualise here!
```

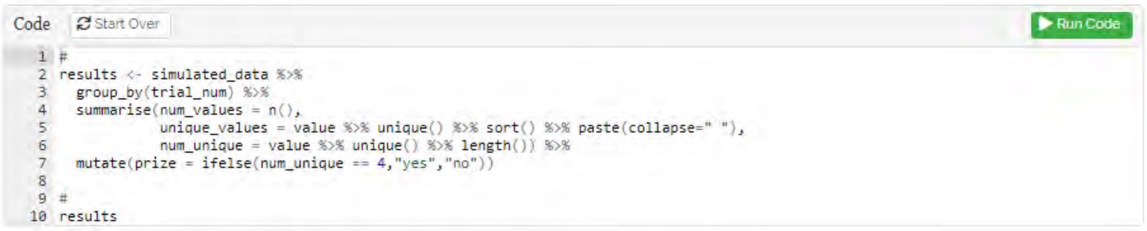
Step 3

Note: You can scroll back up to see previous steps.



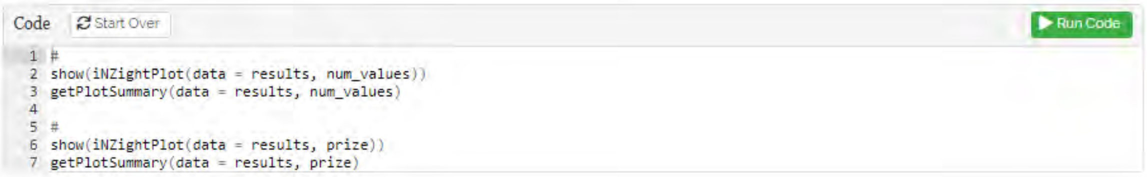
```
#
simulated_data <- generate_data(outcomes,
              probs,
              replace,
              stop_rules,
              num_trials = 100)

# view the data table
simulated_data
```

Step 4



```
#
results <- simulated_data %>%
  group_by(trial_num) %>%
  summarise(num_values = n(),
            unique_values = value %>% unique() %>% sort() %>% paste(collapse=" "),
            num_unique = value %>% unique() %>% length()) %>%
  mutate(prize = ifelse(num_unique == 4,"yes","no"))

#
results
```

Step 5



```
#
show(iNZightPlot(data = results, num_values))
getPlotSummary(data = results, num_values)

#
show(iNZightPlot(data = results, prize))
getPlotSummary(data = results, prize)
```

Mini challenge!

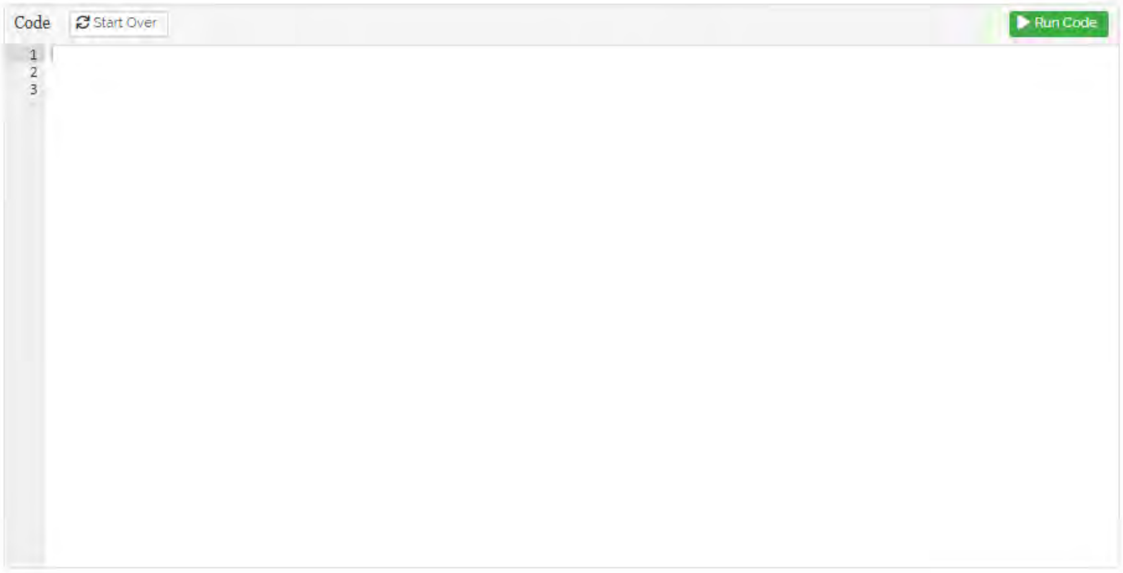
Ask Anna or Maxine for the modified NZQA exemplar task “Oh what a seed”.

Familiarise yourself with the requirements of the task.

Swap control of the computer!

Then try to re-use the code chunks supplied above for the “Oh what a seed” task.

Use the code box below.



Part 5

You have been given all the code provided for the fruity freezes task.

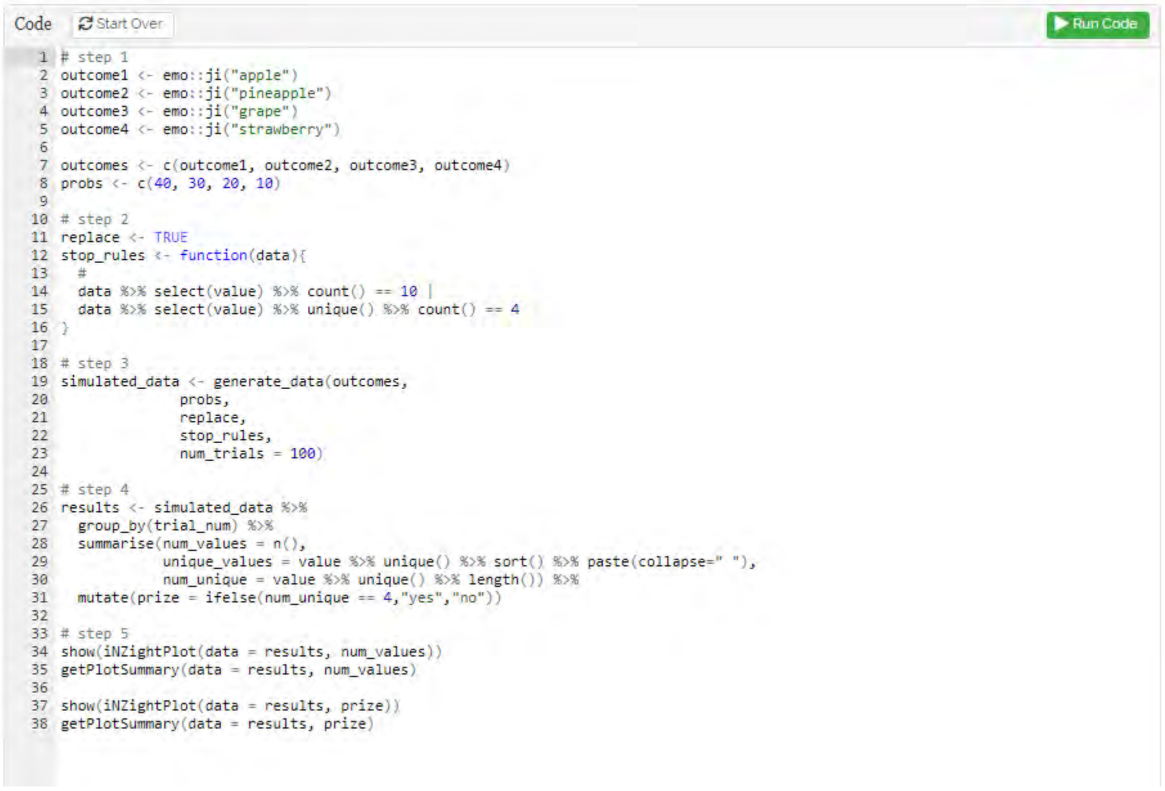
Have a go at changing the code in response to these questions:

What if Grace only buys iceblocks for five days?

What if there was no limit on how many days Grace buys iceblocks for?

How does Grace's chances of winning a prize change as the number of days she buys iceblocks for increases?

Try asking your own "What if" questions and then change the code to see what happens!



```r
# step 1
outcome1 <- emo::ji("apple")
outcome2 <- emo::ji("pineapple")
outcome3 <- emo::ji("grape")
outcome4 <- emo::ji("strawberry")

outcomes <- c(outcome1, outcome2, outcome3, outcome4)
probs <- c(40, 30, 20, 10)

# step 2
replace <- TRUE
stop_rules <- function(data){
  #
  data %>% select(value) %>% count() == 10 |
  data %>% select(value) %>% unique() %>% count() == 4
}

# step 3
simulated_data <- generate_data(outcomes,
              probs,
              replace,
              stop_rules,
              num_trials = 100)

# step 4
results <- simulated_data %>%
  group_by(trial_num) %>%
  summarise(num_values = n(),
            unique_values = value %>% unique() %>% sort() %>% paste(collapse=" "),
            num_unique = value %>% unique() %>% length()) %>%
  mutate(prize = ifelse(num_unique == 4,"yes","no"))

# step 5
show(iNZightPlot(data = results, num_values))
getPlotSummary(data = results, num_values)

show(iNZightPlot(data = results, prize))
getPlotSummary(data = results, prize)
```

Challenge!

If there is time, we'll have a go at this challenge.

Create a new version of a competition involving collecting "cards" (or equivalent, e.g. tiles or figurines, be imaginative!) to win a prize.

Select an existing company (or create a fictional one) that will run this competition.

The company that will run this competition is not sure how many "cards" they want: somewhere between six to ten different "cards".

They want two prizes for the competition – one that over 90% of customers could get if they collected 20 cards and one that no more than 20% of customers could be expected to get if they collected 20 cards.

Use the code provided below to explore different possibilities for a new competition.

You will need to determine how many "cards" to use, how the cards will be "weighted" (i.e. the probabilities of each type of card), and how the two different prizes will be determined.

CODAP links

Link to the model set up: https://codap.concord.org/releases/latest/static/dg/en/cert/index.html#shared=134988

Link to the data table set up: https://codap.concord.org/releases/latest/static/dg/en/cert/index.html#shared=134991

References

Abbasnasab Sardareh, S., Brown, G. T. L., & Denny, P. (2021). Comparing four contemporary statistical software tools for introductory data science and statistics in the social sciences. *Teaching Statistics*, *43*, S157–S172. https://doi.org/10.1111/test.12274

Abrahamson, D., Berland, M., Shapiro, B., Unterman, J., & Wilensky, U. (2006). Leveraging epistemological diversity through computer-based argumentation in the domain of probability. *For the Learning of Mathematics*, *26*(3), 19–45.

Ainley, J., Pratt, D., & Hansen, A. (2006). Connecting engagement and focus in pedagogic task design. *British Educational Research Journal*, *32*(1), 23–38. https://doi.org/10.1080/01411920500401971

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., & Iannone, R. (2021). rmarkdown: Dynamic documents for R. *RStudio*. https://rmarkdown.rstudio.com

American Statistical Association. (2014). *Curriculum guidelines for undergraduate programs in statistical science*. ASA. https://www.amstat.org/docs/default–source/amstat–documents/edu-guidelines2014-11-15.pdf

Anderson, T., & Shattuck, J. (2012). Design-based research: A decade of progress in education research?. *Educational Researcher, 41*(1), 16–25. https://doi.org/10.3102%2F0013189X11428813

Ärlebäck, J. B., Doerr, H. M., & O'Neil, A. H. (2013). A modeling perspective on interpreting rates of change in context. *Mathematical thinking and learning*, *15*(4), 314–336. https://doi.org/10.1080/10986065.2013.834405

Arnold, P. (2013). *Statistical investigative questions. An enquiry into posing and answering investigative questions from existing data* (Doctoral thesis). University of Auckland.

Arnold, P., & Pfannkuch, M. (2016). The language of shape. In D. Ben-Zvi & K. Makar (Eds.), *The teaching and learning of statistics* (pp. 51–61). Springer. https://doi.org/10.1007/978-3-319-23470-0_5

Arnold, P., Confrey, J., Jones, R. S., Lee, H. S., & Pfannkuch, M. (2018). Statistics learning trajectories. In D. Ben-Zvi, K. Makar, & J. Garfield (Eds.), *International handbook of research in statistics education* (pp. 295–326). Springer. https://doi.org/10.1007/978-3-319-66195-7_9

Arnold, P., Pfannkuch, M., Wild, C. J., Regan, M., & Budgett, S. (2011). Enhancing students' inferential reasoning: from hands-on to "movies". *Journal of Statistics Education*, *19*(2). https://doi.org/10.1080/10691898.2011.11889609

Artigue, M. (2002). Learning mathematics in a CAS environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work. *International Journal of Computers for Mathematical Learning*, *7*(3), 245–274. https://doi.org/10.1023/A:1022103903080

Bakker, A. (2018). *Design research in education: A practical guide for early career researchers.* Routledge. https://doi.org/10.4324/9780203701010

Bakker, A., & Gravemeijer, K. (2004). Learning to reason about distribution. In D. Ben-Zvi & J. Garfield (Eds.), *The challenge of developing statistical literacy, reasoning, and thinking* (pp. 147–168). Springer. https://doi.org/10.1007/1-4020-2278-6_7

Bakker, A., & van Eerde, D. (2014). An introduction to design-based research with an example from statistics education. In A. Bikner-Ahsbahs, C. Knipping, & N. Presmeg (Eds.), *Approaches to qualitative research in mathematics education* (pp. 429–466). Springer. https://doi.org/10.1007/978-94-017-9181-6_16

Bargagliotti, A., & Franklin, C. (2015). The statistical education of teachers: Preparing teachers to teach statistics. *CHANCE*, *28*(3), 19–27.

Bargagliotti, A., & Groth, R. (2016). When mathematics and statistics collide in assessment tasks. *Teaching Statistics, 38*(2), 50–55. https://doi.org/10.1111/test.12096

Bargagliotti, A., Franklin, C., Arnold, P., Gould, R., Johnson, S., Perez, L., & Spangler, D. (2020). *Pre-K-12 Guidelines for Assessment and Instruction in Statistics Education (GAISE) report II.* American Statistical Association. https://www.amstat.org/asa/files/pdfs/GAISE/GAISEIIPreK-12_Full.pdf

Baumer, B. S., Kaplan, D. T., & Horton, N. J. (2021). Modern data science with R. Chapman & Hall. https://doi.org/10.1201/9780429200717

Beckman, M. D., delMas, R. C., & Garfield, J. (2017). Cognitive transfer outcomes for a simulation-based introductory statistics curriculum. *Statistics Education Research Journal*, *16*(2), 419–440. https://doi.org/10.52041/serj.v16i2.199

Begg, A. (2004). Statistics curriculum and development: New ways of working. In G. Burrill, M. Camden, & G. Breaux (Eds.), *Curricular development in statistics education, International Association for Statistical Education (IASE) Roundtable, Lund, Sweden, 28 June - 3 July 2004* (pp. 10–20). International Statistical Institute.

Bell, T., & Roberts, J. (2016). Computational thinking is more about humans than computers. *SET*, *1*, 3–7.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20–29.

Ben-Zvi, D. (2000). Toward understanding the role of technological tools in statistical learning. *Mathematical Thinking and Learning*, *2*(1-2), 127–155. https://doi.org/10.1207/S15327833MTL0202_6

Ben-Zvi, D., & Garfield, J. B. (2004). Statistical Literacy, Reasoning and Thinking: Goals, definitions, and challenges. In D. Ben-Zvi, & J. B. Garfield (Eds.), *The Challenge of Developing Statistical Literacy, Reasoning, and Thinking* (pp. 3–16). Springer. https://doi.org/10.1007/1-4020-2278-6_1

Ben-Zvi, D., Gravemeijer, K., & Ainley, J. (2017). Design of statistics learning environments. In D. Ben-Zvi, K. Makar, & J. Garfield (Eds.), *International handbook of research in statistics education* (pp. 473–502). Springer. https://doi.org/10.1007/978-3-319-66195-7_16

Ben-Zvi, D., Makar, K., & Garfield, J. (Eds.). (2018). *International handbook of research in statistics education.* Springer. https://doi.org/10.1007/978-3-319-66195-7

Biehler, R. (1991). Computers in probability education. In R. Kapadia & M. Borovcnik (Eds.), *Chance encounters: Probability in education* (pp. 169–211). Springer. https://doi.org/10.1007/978-94-011-3532-0_6

Biehler, R. (1997a). Software for learning and for doing statistics. *International Statistical Review*, *65*(2), 167–189. https://doi.org/10.1111/j.1751-5823.1997.tb00399.x

Biehler, R. (1997b). Students' difficulties in practicing computer-supported data analysis: Some hypothetical generalizations from results of two exploratory studies. In J. Garfield & G. Burrill (Eds.), *Research on the role of technology in teaching and learning statistics, International Association for Statistical Education Round Table Conference, Granada 1996* (pp. 169–190). International Statistical Institute.

Biehler, R. (2018). Design principles, realizations and uses of software supporting the learning and the doing of statistics - a reflection on developments since the late 1990s. In M. A. Sorto, A. White, & L. Guyot (Eds.), *Looking back, looking forward. Proceedings of the Tenth International Conference on Teaching Statistics (ICOTS10, July, 2018), Kyoto, Japan.* International Statistical Institute.

Biehler, R., & Fleischer, Y. (2021). Introducing students to machine learning with decision trees using CODAP and Jupyter Notebooks. *Teaching Statistics*, *43*, S133–S142. https://doi.org/10.1111/test.12279

Biehler, R., & Schulte, C. (2017). Perspectives for an interdisciplinary data science curriculum at German secondary schools. In R. Biehler, L. Budde, D. Frischemeier, B. Heinemann, S. Podworny, C. Schulte, & T. Wassong (Eds.), *Paderborn Symposium on Data Science Education at School Level 2017: The Collected Extended Abstracts* (pp. 2–14). Universitätsbibliothek Paderborn. http://doi.org/10.17619/UNIPB/1-374

Boehm, F. J. & Hanlon, B. M. (2021). What is happening on Twitter? A framework for student research projects with tweets. *Journal of Statistics and Data Science Education*, *29*, S95–S102. https://doi.org/10.1080/10691898.2020.1848486

Boels, L., Bakker, A., Van Dooren, W., & Drijvers, P. (2019). Conceptual difficulties when interpreting histograms: A review. *Educational Research Review*, *28*(100291). https://doi.org/10.1016/j.edurev.2019.100291

Bootstrap Data Science. (n.d.). *Bootstrap Data Science.* https://bootstrapworld.org/materials/data-science/

Braun, V., & Clark, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101.

Bray, A., Ismay, C., Baumer, B., & Cetinkaya-Rundel, M. (2018). *Infer: Tidy statistical inference.* Retrieved fromhttps://CRAN.R-project.org/package=infer

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, *16*(3), 199–231.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association (April 2012), Vol. 1, Vancouver, Canada.* AERA.

Budgett, S., Pfannkuch, M., Regan, M., & Wild, C. J. (2013). Dynamic visualizations and the randomization test. *Technology Innovations in Statistics Education*, *7*(2). https://doi.org/10.5070/T572013889

Burr, W., Chevalier, F., Collins, C., Gibbs, A. L., Ng, R., & Wild, C. J. (2021). Computational skills by stealth in introductory data science teaching. *Teaching Statistics*, *43*, S34–S51. https://doi.org/10.1111/test.12277

Burrill, G. (2017). Designing interactive dynamic technology activities to support the development of conceptual understanding. In A. Leung, A. & Baccaglini-Frank (Eds.), *Digital technologies in designing mathematics education tasks* (pp. 303–328). Springer. https://doi.org/10.1007/978-3-319-43423-0_15

Burton, N., Brundrett, M., & Jones, M. (2014). *Doing your education research project.* SAGE.

Cambridge Dictionary. (n.d.). Integrate. In *Cambridge.org dictionary.* Retrieved January 18, 2019, from https://dictionary.cambridge.org/dictionary/english/integrate

Case, C., & Jacobbe, T. (2018). A framework to characterize student difficulties in learning inference from a simulation-based approach. *Statistics Education Research Journal*, *17*(2), 9–29. https://doi.org/10.52041/serj.v17i2.156

Case, C., Battles, M., & Jacobbe, T. (2019). Toward an understanding of p-values: Simulation-based inference in a traditional statistics course. *Investigations in Mathematics Learning*, *11*(3), 195-206. https://doi.org/10.1080/19477503.2018.1438869

Casey, S. A., & Wasserman, N. H. (2015). Teachers' knowledge about informal line of best fit. *Statistics Education Research Journal*, *14*(1), 8–35. https://doi.org/10.52041/serj.v14i1.267

Cetinkaya-Rundel, M., & Rundel, C. (2018). Infrastructure and tools for teaching computing throughout the statistical curriculum. *The American Statistician*, *72*(1), 58–65. https://doi.org/10.1080/00031305.2017.1397549

Chamberlin, S. A., & Moon, S. M. (2005). Model-eliciting activities as a tool to develop and identify creatively gifted mathematicians. *Journal of Secondary Gifted Education*, *17*(1), 37–47. https://doi.org/10.4219/jsge-2005-393

Chance, B., & Rossman, A. (2006). Using simulation to teach and learn statistics. In A. Rossman & B. Chance (Eds.), *Working cooperatively in statistics education. Proceedings of the Seventh International Conference on Teaching Statistics (ICOTS7, July 2006), Salvador, Bahia, Brazil.* International Statistics Institute.

Chance, B., Ben-Zvi, D., Garfield, J., & Medina, E. (2007). The role of technology in improving student learning of statistics. *Technology Innovations in Statistics Education*, *1*(1). https://doi.org/10.5070/T511000026

Chance, B., delMas, R., & Garfield, J. (2004). Reasoning about sampling distributions. In D. Ben-Zvi & J. B. Garfield (Eds.), *The challenge of developing statistical literacy, reasoning and thinking* (pp. 295–323). Springer. https://doi.org/10.1007/1-4020-2278-6

Chance, B., Wong, J., & Tintle, N. (2016). Student performance in curricula centered on simulation-based inference: A preliminary report. *Journal of Statistics Education*, *24*(3), 114-126. https://doi.org/10.1080/10691898.2016.1223529

Chance, B.L. (2002). Components of statistical thinking and implications for instruction and assessment. *Journal of Statistics Education, 10*(3). https://doi.org/10.1080/10691898.2002.11910677

Chaput, B., Girard, J. C., & Henry, M. (2008). Modeling and simulations in statistics education. In C. Batanero, G. Burrill, C. Reading & A. Rossman (Eds.), *Joint ICMI/IASE Study: Teaching Statistics in School Mathematics. Challenges for Teaching and Teacher Education.* Proceedings of the ICMI Study 18 and 2008 IASE Round Table Conference. ICMI/IASE.

Clark, J. M., & Paivio, A. (1991). Dual coding theory and education. *Educational Psychology Review*, *3*(3), 149–210. https://doi.org/10.1007/BF01320076

Cleveland, W. S. (2001). Data science: an action plan for expanding the technical areas of the field of statistics. *International statistical review*, *69*(1), 21–26. https://doi.org/10.1111/j.1751-5823.2001.tb00477.x

Cobb, G. (1997). Mere literacy is not enough. In L. Steen (Ed.), *Why numbers count: Quantitative literacy for tomorrow's America* (pp. 75–90). College Entrance Examination Board.

Cobb, G. (2015a). Mere renovation is too little too late: We need to rethink our undergraduate curriculum from the ground up. *The American Statistician*, *69*(4), 266–282. https://doi.org/10.1080/00031305.2015.1093029

Cobb, G. (2015b). Rejoinder to Discussion of "Mere renovation is too little too late: We need to rethink our undergraduate curriculum from the ground up." *The American Statistician*, *69*(4), 266–282. https://doi.org/10.1080/00031305.2015.1093029

Cobb, G. W. (2007). The introductory statistics course: A ptolemaic curriculum? *Technology Innovations in Statistics Education*, *1*(1). https://doi.org/10.5070/T511000028

Cobb, G. W. & Moore, D. S. (1997). Mathematics, statistics, and teaching. *The American Mathematical Monthly, 104*(9), 801–823. https://doi.org/10.1080/00029890.1997.11990723

Cobb, P., & McClain, K. (2004). Principles of instructional design for supporting the development of students' statistical reasoning. In D. Ben-Zvi, & J. B. Garfield (Eds.), *The challenge of developing statistical literacy, reasoning and thinking* (pp. 375–395). Springer. https://doi.org/10.1007/1-4020-2278-6_16

Cook, S. W., & Goldin-Meadow, S. (2006). The role of gesture in learning: Do children use their hands to change their minds? *Journal of Cognition and Development*, *7*(2), 211–232. https://doi.org/10.1207/s15327647jcd0702_4

Creswell, J. W. (2012). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research* (4th ed.). Pearson.

Creswell, J. W., & Plano Clark, V. L. (2011). *Designing and conducting mixed methods research* (2nd ed.). SAGE.

Creswell, J. W., & Poth, C. N. (2016). *Qualitative inquiry and research design: Choosing among five approaches.* SAGE.

Cunningham, K. I. (2021). *Purpose-first programming: A programming learning approach for learners who care most about what code achieves* (PhD dissertation). University of Michigan.

Cutts, Q., Esper, S., Fecho, M., Foster, S. R., & Simon, B. (2012, September). The abstraction transition taxonomy: Developing desired learning outcomes through the lens of situated cognition. In *Proceedings of the ninth annual international conference on International Computing Education Research* (pp. 63–70). https://doi.org/10.1145/2361276.2361290

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2006). Studying aesthetics in photographic images using a computational approach. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *European Conference on Computer Vision* (pp. 288–301). Springer. https://doi.org/10.1007/11744078_23

De Veaux, D., & Velleman, P. F. (2015). Teaching statistics algorithmically or stochastically misses the point: Why not teach holistically?. Online discussion of Cobb, G.W. (2015), "Mere renovation is too little too late: We need to rethink our undergraduate curriculum from the ground up". *The American Statistician*, *69*, 266–282. https://doi.org/10.1080/00031305.2015.1093029

De Veaux, R. D., Agarwal, M., Averett, M., Baumer, B. S., Bray, A., Bressoud, T. C., Bryant, L., Cheng, L. Z., Francis, A., Gould, R., Kim, A. Y., Kretchmar, M., Lu, Q., Moskol, A., Nolan, D., Pelayo, R., Raleigh, S., Sethi, R. J., Sondjaja, M., … Ye, P. (2017). Curriculum guidelines for undergraduate programs in data science. *Annual Review of Statistics and Its Application*, *4*, 15–30. https://doi.org/10.1146/annurev-statistics-060116-053930

Deitrick, E., Wilkerson, M. H., & Simoneau, E. (2017). Understanding student collaboration in interdisciplinary computing activities. In *Proceedings of the 13th annual ACM Conference on International Computing Education Research, Tacoma, WA, USA* (pp. 118–126). Association for Computing Machinery.

delMas, R. (1997). A framework for the evaluation of software for teaching statistical concepts. In J. B. Garfield & G. Burrill (Eds.), *Research on the role of technology in teaching and learning statistics* (pp. 75–90). International Statistical Institute.

Denzin, N. K., & Lincoln, Y. S. (Eds.). (2011). *The Sage handbook of qualitative research.* SAGE.

Dick, T. P., & Burrill, G. F. (2016). Design and implementation principles for dynamic interactive mathematics technologies. In M. Niess, S. Driskell, & K. Hollebrands (Eds.), *Handbook of research on transforming mathematics teacher education in the digital age* (pp. 23–51). IGI Global. https://doi.org/10.4018/978-1-5225-0120-6.ch002

DiSessa, A. A., & Cobb, P. (2004). Ontological innovation and the role of theory in design experiments. *The Journal of the Learning Sciences*, *13*(1), 77–103. https://doi.org/10.1207/s15327809jls1301_4

Doerr, H. M., & Pratt, D. (2008). The learning of mathematics and mathematical modeling. In M. K. Heid & G. W. Blume (Eds.), *Research on technology and the teaching and learning of mathematics: Volume 1 Research syntheses* (pp. 259–285). Information Age Publishing.

Edelson, D. C. (2002). Design research: What we learn when we engage in design. *The Journal of the Learning Sciences*, *11*(1), 105–121. https://doi.org/10.1207/S15327809JLS1101_4

Elliott, T., Soh, Y. H., & Barnett, D. (2021). iNZightPlots: Graphical Tools for Exploring Data with 'iNZight'. R package version 2.13.0. https://CRAN.R-project.org/package=iNZightPlots

Engel, J. (2010). On teaching bootstrap confidence intervals. In C. Reading (Ed.), *Data and context in statistics education: Towards an evidence-based society. Proceedings of the Eighth International Conference on Teaching Statistics (ICOTS8, July 2010), Ljubljana, Slovenia.* International Statistical Institute.

Engel, J. (2017). Statistical literacy for active citizenship: A call for data science education. *Statistics Education Research Journal, 16* (1), 44–49. https://doi.org/10.52041/serj.v16i1.213

Engel, J., Erickson, T., & Martignon, L. (2019). Teaching about decision trees for classification problems. In S Budgett (Ed.), *Decision making based on data. Proceedings of the satellite conference of the International Association for Statistical Education (IASE), Kuala Lumpur, Malaysia.* IASE.

Erickson, T. (2006). Using simulation to learn about inference. In A. Rossman & B. Chance (Eds.), *Working cooperatively in statistics education. Proceedings of the Seventh International Conference on Teaching Statistics (ICOTS7, July 2006), Salvador, Bahia, Brazil.* International Statistics Institute.

Erickson, T. (2013). Designing games for understanding in a data analysis environment. *Technology Innovations in Statistics Education*, *7(*2). https://doi.org/10.5070/T572013897

Erickson, T. (2020). The BART data portal. *An introduction to data science with CODAP.* http://codap.xyz/awash/bart-chapter.html

Erickson, T., Wilkerson, M., Finzer, B., & Reichsman, F. (2019). Data moves. *Technology Innovations in Statistics Education*, *12*(1). https://doi.org/10.5070/T5121038001

Fergusson, A. (2017). *Informally testing the fit of a probability distribution model* (Masters dissertation). University of Auckland.

Fergusson, A., & Bolton, E. L. (2018). Exploring modern data in a large introductory statistics course. In M. A. Sorto, A. White, & L. Guyot (Eds.), *Looking back, looking forward. Proceedings of the Tenth International Conference on Teaching Statistics (ICOTS10, July, 2018), Kyoto, Japan.* International Statistical Institute.

Fergusson, A., & Pfannkuch, M. (2020). Development of an informal test for the fit of a probability distribution model for teaching. *Journal of Statistics Education*, *28*(3), 344–357. https://doi.org/10.1080/10691898.2020.1837039

Fergusson, A., & Pfannkuch, M. (2021). Introducing teachers who use GUI-driven tools for the randomization test to code-driven tools. *Mathematical Thinking and Learning.* https://doi.org/10.1080/10986065.2021.1922856

Fergusson, A., & Pfannkuch, M. (2022). Introducing high school statistics teachers to predictive modelling and APIs using code-driven tools. *Statistics Education Research Journal*, *21*(2), Article 8. https://doi.org/10.52041/serj.v21i2.49

Fergusson, A., & Wild, C. J. (2021). On traversing the data landscape: Introducing APIs to data-science students. *Teaching Statistics*, *43*, S71–S83. https://doi.org/10.1111/test.12266

Ferreira, R. D. S., Kataoka, V. Y., & Karrer, M. (2014). Teaching probability with the support of the R statistical software. *Statistics Education Research Journal*, *13*(2), 132–147. https://doi.org/10.52041/serj.v13i2.286

Finch, S., Gordon, I., & Patrick, C. (2021). Taking the aRghhhh out of teaching statistics with R: Using R Markdown. *Teaching Statistics*, *43*, S143–S147. https://doi.org/10.1111/test.12251

Finzer, W. (2013). The data science education dilemma. *Technology Innovations in Statistics Education*, *7*(2). https://doi.org/10.5070/T572013891

Finzer, W. (2016). *Common online data analysis platform (CODAP)*. The Concord Consortium. https://codap.concord.org/

Finzer, W., & Reichsman, F. (2018). Exploring the essential elements of data science education. https://concord.org/newsletter/2018-fall/exploring-the-essential-elements-of-data-science-education/

Fleischer, Y., Biehler, R., & Schulte, C. (2022). Teaching and learning data-driven machine learning with educationally designed Jupyter notebooks. *Statistics Education Research Journal*, *21*(2). https://doi.org/10.52041/serj.v21i2.61

Forbes, S., Chapman, J., Harraway, J., Stirling, D., & Wild, C. J. (2014). Use of data visualisation in the teaching of statistics: A New Zealand perspective. *Statistics Education Research Journal*, *13*(2). https://doi.org/10.52041/serj.v13i2.290

Friel, S. N., Curcio, F. R., & Bright, G. W. (2001). Making sense of graphs: Critical factors influencing comprehension and instructional implications. *Journal for Research in Mathematics Education*, *32*(2), 124–158. https://doi.org/10.2307/749671

Fries, L., Son, J. Y., Givvin, K. B., & Stigler, J. W. (2021). Practicing connections: A framework to guide instructional design for developing understanding in complex domains. *Educational Psychology Review*, *33*(2), 739–762. https://doi.org/10.1007/s10648-020-09561-x

Gafny, R. & Ben-Zvi, D. (2021). Middle school students' articulations of uncertainty in non-traditional big data IMA learning environments. In *Proceedings from the 12th International Collaboration for Research on Statistical Reasoning, Thinking and Literacy*, *Virtual* (pp. 40–43). SRTL.

Garfield, J., & Ben-Zvi, D. (2009). Helping students develop statistical reasoning: Implementing a statistical reasoning learning environment. *Teaching Statistics*, *31*(3), 72–77. https://doi.org/10.1111/j.1467-9639.2009.00363.x

Garfield, J., & DelMas, R. (2010). A web site that provides resources for assessing students' statistical literacy, reasoning and thinking. *Teaching Statistics*, *32*(1), 2–7. https://doi.org/10.1111/j.1467-9639.2009.00373.x

Garfield, J., delMas, R., & Zieffler, A. (2010). Developing tertiary-level students' statistical thinking through the use of model-eliciting activities. In C. Reading (Ed.), *Data and context in statistics education: Towards an evidence-based society. Proceedings of the Eighth International Conference on Teaching Statistics (ICOTS8, July, 2010), Ljubljana, Slovenia.* International Statistical Institute.

Garfield, J., delMas, R., & Zieffler, A. (2012). Developing statistical modelers and thinkers in an introductory, tertiary-level statistics course. *ZDM*, *44*(7), 883–898. https://doi.org/10.1007/s11858-012-0447-5

Gelman, A., & Nolan, D. (2017). *Teaching statistics: A bag of tricks.* Oxford University Press.

Gigerenzer, G. (1998). We need statistical thinking, not statistical rituals. *Behavioral and Brain Sciences*, *21*(2), 199–200. https://doi.org/10.1017/S0140525X98281167

Gould, R. (2010). Statistics and the modern student. *International Statistical Review*, *78*(2), 297–315. https://doi.org/10.1111/j.1751-5823.2010.00117.x

Gould, R. (2015). *Data science for all: A glimpse into the future.* Keynote presented at the New Zealand Association of Mathematics Teachers conference, Auckland, New Zealand.

Gould, R. (2017). Data literacy is statistical literacy. *Statistics Education Research Journal, 16*(1), 22–25. https://doi.org/10.52041/serj.v16i1.209

Gould, R. (2021). Toward data-scientific thinking. *Teaching Statistics, 43,* S11–S22. https://doi.org/10.1111/test.12267

Gould, R., Davis, G., Patel, R., & Esfandiari, M. (2010). Enhancing conceptual understanding with data driven labs. In C. Reading (Ed.), *Data and context in statistics education: Towards an evidence-based society. Proceedings of the Eighth International Conference on Teaching Statistics (ICOTS8, July 2010), Ljubljana, Slovenia.* International Statistical Institute.

Gould, R., Moncada-Machado, S., Johnson, T. A., & Molyneux, J. (2017). *Introduction to Data Science Curriculum* (version 4). UCLA/LAUD.

Gould, R., Peng, R. D., Kreuter, F., Pruim, R., Witmer, J., & Cobb, G. W. (2018). Challenge to the established curriculum: A collection of reflections. In D. Ben-Zvi, K. Makar, & J. Garfield (Eds.), *International handbook of research in statistics education* (pp. 415–432). Springer. https://doi.org/10.1007/978-3-319-66195-7_13

Grant, R. (2017). Statistical literacy in the data science workplace. *Statistics Education Research Journal, 16*(1). https://doi.org/10.52041/serj.v16i1.207

Gravemeijer, K. (2004). Local instruction theories as a means of support for teachers in reform mathematics education. *Mathematical Thinking and Learning*, *6*(2), 105–128. https://doi.org/10.1207/s15327833mtl0602_3

Grolemund, G., & Wickham, H. (2014). A cognitive interpretation of data analysis. *International Statistical Review*, *82*(2), 184–204.

Grover, S., Jackiw, N., & Lundh, P. (2019). Concepts before coding: Non-programming interactives to advance learning of introductory programming concepts in middle school. *Computer Science Education*, *29*(2-3), 106–135. https://doi.org/10.1080/08993408.2019.1568955

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, *25*(2), 199-237. https://doi.org/10.1080/08993408.2015.1033142

Hammond, M., & Wellington, J. (2013). *Research methods: The key concepts.* Routledge.

Hardin, J. (2018). Dynamic data in the statistics classroom. *Technology Innovations in Statistics Education, 11*(1). https://doi.org/10.5070/T5111031079

Hardin, J., Hoerl, R., Horton, N. J., Nolan, D., Baumer, B., Hall-Holt, O., Murrell, P., Peng, R., Roback, P., Temple Lang, D. & Ward, M. D. (2015). Data science in statistics curricula: Preparing students to "think with data". *The American Statistician*, *69*(4), 343–353. https://doi.org/10.1080/00031305.2015.1077729

Hermans, F., Swidan, A., & Aivaloglou, E. (2018, May). Code phonology: An exploration into the vocalization of code. In *Proceedings of the 26th Conference on Program Comprehension* (pp. 308–311). https://doi.org/10.1145/3196321.3196355

Hesterberg, T. C. (1998). Simulation and bootstrapping for teaching statistics. In *American Statistical Association Proceedings of the section on statistical education* (pp. 44–52). ASA.

Hicks, S. C., & Irizarry, R. A. (2018). A guide to teaching data science. *The American Statistician*, *72*(4), 382–391. https://doi.org/10.1080/00031305.2017.1356747

Hipkins, R. (2014). *Doing research that matters: A success story from statistics education.* New Zealand Council for Educational Research.

Hjalmarson, M. A., Moore, T. J., & delMas, R. (2011). Statistical analysis when the data is an image: Eliciting student thinking about sampling and variability. *Statistics Education Research Journal, 10*(1). https://doi.org/10.52041/serj.v10i1.353

Hoadley, C., & Campos, F. C. (2022). Design-based research: What it is and why it matters to studying online learning. *Educational Psychologist*, *57*(3), 207–220. https://doi.org/10.1080/00461520.2022.2079128

Hoerl, R. W., Snee, R. D., & De Veaux, R. D. (2014). Applying statistical thinking to 'Big Data' problems. *Wiley Interdisciplinary Reviews: Computational Statistics*, *6*(4), 222–232. https://doi.org/10.1002/wics.1306

Hoover, A. K., Barnes, J., Fatehi, B., Moreno-León, J., Puttick, G., Tucker-Raymond, E., & Harteveld, C. (2016, October). Assessing computational thinking in students' game designs. In *Proceedings of the 2016 annual symposium on computer-human interaction in play companion extended abstracts (October 2016), Austin, Texas, USA* (pp. 173–179). ACM. https://doi.org/10.1145/2968120.2987750

Horton, N. J., Baumer, B. S., & Wickham, H. (2014). Teaching precursors to data science in introductory and second courses in statistics. In K. Makar, B. de Sousa, & R. Gould (Eds.), *Sustainability in Statistics Education. Proceedings of the Ninth International Conference on Teaching Statistics (ICOTS9, July, 2014)*, Flagstaff, Arizona, USA. International Statistical Institute.

Horton, N. J., Chao, J., Finzer, W., & Palmer, P. (2022). Spam four ways: Making sense of text data. *CHANCE*, *35*(2), 32–40. https://doi.org/10.1080/09332480.2022.2066414

Horton, N., Chao, J., & Finzer, W. (2021). How learners produce data from text in classifying clickbait. In *Proceedings from the 12th International Collaboration for Research on Statistical Reasoning, Thinking and Literacy*, *Virtual* (pp. 36–39). SRTL.

Introduction to Data Science. (n.d.). *Introduction to Data Science (IDS).* https://www.idsucla.org

International Data Science in Schools Project. (n.d.). International Data Science in Schools Project (IDSSP). http://www.idssp.org/pages/framework.html

Jacobbe, T., Whitaker, D., Case, C., & Foti, S. (2014). The LOCUS assessment at the college level: Conceptual understanding in introductory statistics. In K. Makar, B. de Sousa, & R. Gould (Eds.), *Sustainability in Statistics Education. Proceedings of the Ninth International Conference on Teaching Statistics (ICOTS9, July, 2014)*, Flagstaff, Arizona, USA. International Statistical Institute.

Kaplan, D. (2007). Computing and introductory statistics. *Technology Innovations in Statistics Education*, *1*(1). https://doi.org/10.5070/T511000030

Kaplan, D. (2018). Teaching stats for data science. *The American Statistician*, *72*(1), 89–96. <https://doi.org/10.1080/00031305.2017.1398107

Kaplan, J. J., Gabrosek, J. G., Curtiss, P., & Malone, C. (2014a). Investigating student understanding of histograms. *Journal of Statistics Education*, *22*(2). https://doi.org/10.1080/10691898.2014.11889701

Kaplan, J. J., Rogness, N., & Fisher, D. (2014b). Exploiting lexical ambiguity to help students understand the meaning of random. *Statistics Education Research Journal*, *13*(1), 9–24. https://doi.org/10.52041/serj.v13i1.296

Kazak, S., Fujita, T., & Turmo, M. P. (2021). Students' informal statistical inferences through data modeling with a large multivariate dataset. *Mathematical Thinking and Learning*. https://doi.org/10.1080/10986065.2021.1922857

Kim, A. Y., Ismay, C., & Chunn, J. (2018). The fivethirtyeight R Package:'Tame Data' principles for introductory statistics and data science courses. *Technology Innovations in Statistics Education*, *11*(1). https://doi.org/10.5070/T5111035892

Konold, C., & Kazak, S. (2008). Reconnecting data and chance. *Technology Innovations in Statistics Education*, *2*(1). https://doi.org/10.5070/T521000032

Konold, C., & Miller, C. (2015). *TinkerPlots™ Version 2.3 [Computer Software]*. Learn Troop. http://www.tinkerplots.com/

Leavy, A., & Hourigan, M. (2016). Crime scenes and mystery players! Using driving questions to support the development of statistical literacy. *Teaching Statistics*, *38*(1), 29–35. https://doi.org/10.1111/test.12088

Lee, H. S., Doerr, H. M., Tran, D., & Lovett, J. N. (2016). The role of probability in developing learners' models of simulation approaches to inference. *Statistics Education Research Journal*, *15*(2), 216–238. https://doi.org/10.52041/serj.v15i2.249

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

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

Lesh, R. A., Cramer, K., Doerr, H. M., Post, T., & Zawojewski, J. S. (2003). Model development sequences. In R. A. Lesh & H. M. Doerr (Eds.), *Beyond constructivism: Models and modeling perspectives on mathematics problem solving, learning, and teaching* (pp. 35– 58). Routledge.

Lesh, R., & Doerr, H. M. (2000). Symbolizing, communicating, and mathematizing: Key components of models and modeling. In P. Cobb, E. Yackel, & K. McLain (Eds.), *Symbolizing and communicating in mathematics classrooms: Perspectives on discourse, tools, and instructional design* (pp. 361–383). Routledge. https://doi.org/10.4324/9781410605351

Li, J. (2018). Statistical methods for image analysis. http://personal.psu.edu/jol2/Li_lecture_highschool.pdf

Lock, R., Frazer Lock, P., Lock Morgan, K., Lock, E., & Lock, D. (2013). *Statistics: Unlocking the power of data.* Wiley.

Madden, S. (2021). Stochastics and computational thinking: A response to Kazak and Pratt. *Research in Mathematics Education*, *23*(2), 134–141. https://doi.org/10.1080/14794802.2021.1958367

Madden, S. (2018). Impacting mathematical and technological creativity with dynamic technology scaffolding. In V. Freiman & J. Tassell (Eds.), *Creativity and technology in mathematics education*, (pp. 89–124). Springer International Publishing

Magana, A. J., Vasileska, D., & Ahmed, S. (2011). Work in progress—a transparency and scaffolding framework for computational simulation tools. *2011 Frontiers in Education Conference (FIE)*, (pp. S4G-1–S4G-2). IEEE. https://doi.org/10.1109/FIE.2011.6142803

Makar, K., & Rubin, A. (2018). Learning about statistical inference. In D. Ben-Zvi, K. Makar, & J. Garfield (Eds.), *International handbook of research in statistics education* (pp. 261–294). Springer. https://doi.org/10.1007/978-3-319-66195-7_8

Matuszewski, A. L. (2018). High school statistics teachers' understanding of hypothesis testing through simulation (PhD dissertation). Middle Tennessee State University.

Maxara, C., & Biehler, R. (2006). Students' probabilistic simulation and modeling competence after a computer-intensive elementary course in statistics and probability. In A. Rossman & B. Chance (Eds.), *Working cooperatively in statistics education. Proceedings of the Seventh International Conference on Teaching Statistics (ICOTS7, July 2006), Salvador, Bahia, Brazil.* International Statistics Institute.

McKenney, S., & Reeves, T. C. (2018). *Conducting educational design research.* Routledge. https://doi.org/10.4324/9781315105642

McNamara, A. A. (2015). *Bridging the gap between tools for learning and for doing statistics* (PhD dissertation). University of California, Los Angeles.

Ministry of Education. (1992). *Mathematics in the New Zealand curriculum.* Learning Media.

Ministry of Education. (2007). *The New Zealand Curriculum.* Learning Media.

Molyneux, J., McNamara, A., Johnson, T. & Nolen, S. (2017). *mobilizr: Functions for the mobilize introduction to data science course.* R package version 0.3.12.

Moore, D. S. (1997). New pedagogy and new content: The case of statistics. *International Statistical Review, 65*(2), 123–137. https://doi.org/10.1111/j.1751-5823.1997.tb00390.x

Morgan, D. L. (2014a). *Integrating qualitative and quantative methods: A pragmatic approach.* SAGE.

Morgan, D. L. (2014b). Pragmatism as a paradigm for social research. *Qualitative inquiry, 20*(8), 1045–1053. https://doi.org/10.1177/1077800413513733

Myint, L., Hadavand, A., Jager, L. & Leek, J. (2020). Comparison of beginning R students' perceptions of peer-made plots created in two plotting systems: A randomized experiment, *Journal of Statistics Education, 28*(1), 98–108. https://doi.org/10.1080/10691898.2019.1695554

Nardelli, E. (2019). Do we really need computational thinking? *Commun. ACM*, *62*(2), 32–35. https://doi.org/10.1145/3231587

NASEM. (2018). *Data science for undergraduates: Opportunities and options.* The National Academies of Sciences Engineering Medicine. https://www.nap.edu/catalog/25104/data-science-for-undergraduates-opportunities-and-options

Nilsson, P. (2014). Experimentation in probability teaching and learning. In E. Chernoff & B. Sriraman (Eds.), *Probabilistic thinking: Presenting plural perspectives*, (pp. 509–532). Springer. https://doi.org/10.1007/978-94-007-7155-0_28

Nolan, D., & Temple Lang, D. (2010). Computing in the statistics curricula. *The American Statistician*, *64*(2), 97–107. https://doi.org/10.1198/tast.2010.09132

Nolan, D., & Temple Lang, D. (2015). *Data science in R: A case studies approach to computational reasoning and problem solving.* CRC Press. https://doi.org/10.1201/b18325

Noll, J., & Kirin, D. (2016). Student approaches to constructing statistical models using TinkerPlots. *Technology Innovations in Statistics Education*, *9*(1). https://doi.org/10.5070/T591023693

Noll, J., Clement, K., Dolor, J., Kirin, D., & Petersen, M. (2018). Students' use of narrative when constructing statistical models in TinkerPlots. *ZDM*, *50*(7), 1267–1280. https://doi.org/10.1007/s11858-018-0981-x

NZQA. (2019). *Annotated exemplar Level 3 AS91581.* New Zealand Qualifications Authority. https://www.nzqa.govt.nz/ncea/subjects/mathematics/exemplars/level-3-as91581/

Patel, A. (2021). *Statistical Modelling: An enquiry into novice students' co-creation of reasoning and practice* (Doctoral thesis). University of Auckland.

Patel, A., & Pfannkuch, M. (2018). Developing a statistical modeling framework to characterize Year 7 students' reasoning. *ZDM*, *50*(7), 1197–1212. https://doi.org/10.1007/s11858-018-0960-2

Patitsas, E., Berlin, J., Craig, M., & Easterbrook, S. (2019). Evidence that computer science grades are not bimodal. *Communications of the ACM*, *63*(1), 91–98. https://doi.org/10.1145/3372161

Patton, M. Q. (2015). *Qualitative research and evaluation methods* (4th ed.). SAGE.

Peng, R. D., & Matsui, E. (2017). *The art of data science.* Skybrude Consulting, LLC. https://bookdown.org/rdpeng/artofdatascience/

Perrenet, J., Groote, J. F., & Kaasenbrood, E. (2005). Exploring students' understanding of the concept of algorithm: Levels of abstraction. *ACM SIGCSE Bulletin*, *37*(3), 64-68. https://doi.org/10.1145/1151954.1067467

Pfannkuch, M. (2011). The role of context in developing informal statistical inferential reasoning: A classroom study. *Mathematical Thinking and Learning*, *13*(1–2), 27–46. https://doi.org/10.1080/10986065.2011.538302

Pfannkuch, M. (2018). Reimagining curriculum approaches. In D. Ben-Zvi, K. Makar, & J. Garfield (Eds.), *International handbook of research in statistics education* (pp. 387–413). Springer. https://doi.org/10.1007/978-3-319-66195-7_12

Pfannkuch, M., & Budgett, S. (2016). Markov processes: Exploring the use of dynamic visualizations to enhance student understanding. *Journal of Statistics Education*, *24*(2), 63–73. https://doi.org/10.1080/10691898.2016.1207404

Pfannkuch, M., & Ziedins, I. (2014). A modelling perspective on probability. In E. Chernoff & B. Sriraman (Eds.), *Probabilistic thinking: Presenting plural perspectives,* (pp. 101–116). Springer. https://doi.org/10.1007/978-94-007-7155-0_5

Pfannkuch, M., Arnold, P., & Wild, C. W. (2011). *Building students' inferential reasoning: Statistics curriculum levels 5 and 6.*Summary research report for the Teaching and Learning Research Initiative. http://www.tlri.org.nz/sites/default/files/projects/9275-summaryreport_0.pdf

Pfannkuch, M., Budgett, S., Fewster, R., Fitch, M., Pattenwise, S., Wild, C., & Ziedins, I. (2016). Probability modeling and thinking: What can we learn from practice?. *Statistics Education Research Journal*, *15*(2), 11–37. https://doi.org/10.52041/serj.v15i2.238

Pfannkuch, M., Forbes, S., Harraway, J., Budgett, S., & Wild, C. J. (2013). *Bootstrapping students' understanding of statistical inference.* Summary research report for the Teaching and Learning Research Initiative. http://www.tlri.org.nz/sites/default/files/projects/9295_summary%20report.pdf

Podworny, S., & Biehler, R. (2014). A learning trajectory on hypothesis testing with TinkerPlots-Design and exploratory evaluation. In K. Makar, B. de Sousa, & R. Gould (Eds.), *Sustainability in Statistics Education. Proceedings of the Ninth International Conference on Teaching Statistics (ICOTS9, July, 2014)*, Flagstaff, Arizona, USA. International Statistical Institute.

Podworny, S., Fleischer, Y., Hüsing, S., Biehler, R., Frischemeier, D., Höper, L., & Schulte, C. (2021). Using data cards for teaching data based decision trees in middle school. *21st Koli Calling International Conference on Computing Education Research.* https://doi.org/10.1145/3488042.3489966

Pratt, D. (2011). Re-connecting probability and reasoning from data in secondary school teaching. In *Proceedings of the 58th International Statistical Institute World Statistical Congress, Dublin*, (pp. 890–899). International Statistical Institute.

Pratt, D., Jones, I., & Prodromou, T. (2006). An elaboration of the design construct of phenomenalisation. In A. Rossman & B. Chance (Eds.), *Working cooperatively in statistics education. Proceedings of the Seventh International Conference on Teaching Statistics (ICOTS7, July 2006), Salvador, Bahia, Brazil.* International Statistics Institute.

ProDaBi. (n.d.). *ProDaBi.* http://www.prodabi.de

Prodromou, T. (2014). Developing a modelling approach to probability using computer-based simulations. In E. Chernoff, & B. Sriraman (Eds.), *Probabilistic thinking: Presenting plural perspectives,* (pp. 417–439). Springer. https://doi.org/10.1007/978-94-007-7155-0_22

Pruim, R., Kaplan, D. T., & Horton, N. J. (2017). The mosaic package: Helping students to 'think with data' using R. *The R Journal*, *9*(1), 77–102. https://journal.r-project.org/archive/2017/RJ-2017-024/index.html

Punch, K. F., & Oancea, A. (2014). *Introduction to research methods in education* (2nd ed.). SAGE.

R Core Team. (2020). R: A language and environment for statistical computing. https://www.R-project.org

Rabardel, P., & Beguin, P. (2005). Instrument mediated activity: From subject development to anthropocentric design. *Theoretical Issues in Ergonomics Science*, *6*(5), 429–461. https://doi.org/10.1080/14639220500078179

Radonich, P. (2014). *Developing a mechanism for producing follow-up tasks which use the students' methods that have emerged from a Model-Eliciting Activity* (Doctoral thesis). University of Auckland.

realtime. (n.d.). *JamaicaVM 3.4 — user documentation: The virtual machine for realtime and embedded systems.* Retrieved fromhttp://aicas.com/jamaica/3.4/doc/html/x3718.html

Reeves, T. C. (2007). Design-based research from a technology perspective. In J. Van den Akker, K. Gravemeijer, S. McKenney & N. Nieveen (Eds.), *Educational design research*, (pp. 52–56). Routledge.

Repenning, A., Webb, D., & Ioannidou, A. (2010, March). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on computer science education (March 2010), Milwaukee, Wisconsin, USA* (pp. 265–269). ACM. https://doi.org/10.1145/1734263.1734357

Ricketts, C., & Berry, J. (1994). Teaching statistics through resampling. *Teaching Statistics*, *16*(2), 41–44. https://doi.org/10.1111/j.1467-9639.1994.tb00685.x

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

Rossman, A., & Nolan, D. (2015). Interview with Deborah Nolan. *Journal of Statistics Education*, *23*(3). https://doi.org/10.1080/10691898.2015.11889751

RStudio Team. (2018). *RStudio: Integrated development environment for R.* http://www.rstudi o.com/

Sánchez, E. (2002). Teachers' beliefs about usefulness of simulation with the educational software Fathom for developing probability concepts in statistics classroom. In E. Phillips (Ed.), *Proceedings of the Sixth International Conference on Teaching Statistics (ICOTS6, July 2002), Cape Town, South Africa.* International Statistical Institute.

Sandoval, W. (2014). Conjecture mapping: An approach to systematic educational design research. *Journal of the Learning Sciences*, *23*(1), 18–36. https://doi.org/10.1080/10508406.201 3.778204

Schloerke, B., Allaire, J., & Borges, B. (2018). Learnr: Interactive tutorials for R. *CRAN.*https://CRAN.R-project.org/package=learnr

Schulte, C. (2008). Block Model: An educational model of program comprehension as a tool for a scholarly approach to teaching. In *Proceedings of the fourth international workshop on computing education research (September 2008), Sydney, Australia.* (pp. 149–160). ACM. https://doi.org/10.1145/1404520.1404535

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469–495. https://doi.org/10.1007/s10639-016-9482-0

Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: A sociocultural perspective. *Computer Science Education, 29*(2-3), 136–176. https://doi.org/10.1080/08993408.2019.1608781

Shoop, R., Flot, J., Higashi, R., Witherspoon, E., & McKenna, J. (2016). *Using model eliciting activities to engage students in computational thinking practices in robotics classrooms.* Paper presented at the High Impact Technology Exchange Conference (2016 HI-TECH, July 2016), Pittsburgh, Pennsylvania.

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, *15*(2), 4–14. https://doi.org/10.3102/0013189X015002004

Simon, J. L., Atkinson, D. T., & Shevokas, C. (1976). Probability and statistics: Experimental results of a radically different teaching method. *The American Mathematical Monthly*, *83*(9), 733–739. https://doi.org/10.1080/00029890.1976.11994231

Snee, R. D. (1990). Statistical thinking and its contribution to total quality. *The American Statistician*, *44*(2), 116–121. https://doi.org/10.2307/2684144

Soloway, E. M., & Woolf, B. (1980). Problems, plans, and programs. *ACM SIGCSE Bulletin*, *12*(1), 16–24. https://doi.org10.1145/800140.804605

Son, J. Y., Blake, A. B., Fries, L., & Stigler, J. W. (2021). Modeling first: Applying learning science to the teaching of introductory statistics. *Journal of Statistics and Data Science Education*, *29*(1), 4–21. https://doi.org/10.1080/10691898.2020.1844106

Starnes, D., & Martin, A. (2015). The AP statistics exam: An insider's guide to its distinctive features. *CHANCE*, *28*(3), 28–37. https://doi.org/10.1080/09332480.2015.1099363

Stigler, J. W., & Son, J. Y. (2018). Modeling first: A modeling approach to teaching introductory statistics. In M. A. Sorto, A. White, & L. Guyot (Eds.), *Looking back, looking forward. Proceedings of the Tenth International Conference on Teaching Statistics (ICOTS10, July, 2018), Kyoto, Japan.* International Statistical Institute.

Sun, D. (2018). *What should we teach in data science courses?* Webinar presented for C.A.U.S.E. https://www.causeweb.org/cause/webinar/teaching/2018-02-13

Sweller, J., van Merriënboer, J. J. G., & Paas, F. G. W. (1998). Cognitive architecture and instructional design. *Educational Psychology Review*, *10*(3), 251–296. https://doi.org/10.1023/A:1022193728205

Tenenberg, J., & Knobelsdorf, M. (2014). Out of our minds: a review of sociocultural cognition theory. *Computer Science Education*, *24*(1), 1–24. https://doi.org/10.1080/08993408.2013.869396

Thoma, S., Deitrick, E., & Wilkerson, M. (2018). “It didn’t really go very well”: Epistemological framing and the complexity of interdisciplinary computing activities. In J. Kay & R. Luckin (Eds.), *Rethinking learning in digital age: Making the learning sciences count. Proceedings of the 13th International Conference of the Learning Sciences (ICLS)*, *London, UK*, (Vol. 2, pp. 1121–1124). International Society of the Learning Sciences.

Thomas, F., & Moore, J. (1980). CUSUM: Computer simulation for statistics teaching. *Teaching Statistics*, *2*(1), 23–28. https://doi.org/10.1111/j.1467-9639.1980.tb00374.x

Tintle, N. L., Topliff, K., VanderStoep, J., Holmes, V. L., & Swanson, T. (2012). Retention of statistical concepts in a preliminary randomization-based introductory statistics curriculum. *Statistics Education Research Journal*, *11*(1), 21–40. https://doi.org/10.52041/serj.v11i1.340

Tintle, N., Clark, J., Fischer, K., Chance, B., Cobb, G., Roy, S., Swanson, T. & VanderStoep, J. (2018). Assessing the association between precourse metrics of student preparation and student performance in introductory statistics: Rests from early data on simulation-based inference vs. nonsimulation-based inference. *Journal of Statistics Education*, *26*(2), 103–109. https://doi.org/10.1080/10691898.2018.1473061

Tran, D., & Lee, H. S. (2015). Considerations for design and implementation of statistics tasks. In *Teaching statistics through data investigations MOOC-Ed,* Friday Institute for Educational

Innovation, NC State University, Raleigh, NC. http://fi-courses.s3.amazonaws.com/tsdi/unit_3/CDIST.pdf

Trouche, L. (2004). Managing the complexity of human/machine interactions in computerized learning environments: Guiding students' command process through instrumental orchestrations. *International Journal of Computers for Mathematical Learning*, *9*(3), 281–307. https://doi.org/10.1007/s10758-004-3468-5

Utts, J. (2015). Challenges, changes and choices in the undergraduate statistics curriculum. Online discussion of Cobb, G.W. (2015), "Mere renovation is too little too late: We need to rethink our undergraduate curriculum from the ground up". *The American Statistician*, *69*, 266–282. https://doi.org/10.1080/00031305.2015.1093029

Van den Akker, J. (1999). Principles and methods of development research. In J. Van den Akker, R. M. Branch, K. Gustafson, N. Nieveen & T. Plomp (Eds.), *Design approaches and tools in education and training,* (pp. 1–14). Springer. https://doi.org/10.1007/978-94-011-4255-7_1

van Dijke-Droogers, M., Drijvers, P., & Bakker, A. (2021). Statistical modeling processes through the lens of instrumental genesis. *Educational Studies in Mathematics*, *107*(2), 235–260. https://doi.org/10.1007/s10649-020-10023-y

Van Merrienboer, J. J., & Krammer, H. P. (1987). Instructional strategies and tactics for the design of introductory computer programming courses in high school. *Instructional Science*, *16*(3), 251–285. https://doi.org/10.1007/BF00120253

Van Someren, M. W., Barnard, Y. F., & Sandberg, J. A. C. (1994). *The think aloud method: A practical approach to modelling cognitive processes.* Academic Press.

Watkins, A., Bargagliotti, A. E., & Franklin, C. (2014). Simulation of the sampling distribution of the mean can mislead. *Journal of Statistics Education*, *22*(3), 1–21. https://doi.org/10.1080/10691898.2014.11889716

Watson, A., & Ohtani, M. (Eds.). (2015). *Task design in mathematics education.* Springer International Publishing. https://doi.org/10.1007/978-3-319-09629-2

Weaver, K. (2018). Pragmatic paradigm. In B. Frey (Ed.), *The SAGE encyclopedia of educational research, measurement, and evaluation* (Vol. 1, pp. 1287–1288). SAGE. https://dx.doi.org/10.4135/9781506326139.n534

Weiland, T. (2016). The importance of context in task selection. *Teaching Statistics, 39(*1), 20–25. https://doi.org/10.1111/test.12116

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). 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

Wickham, H. (2010). Using visualisation to teaching data analysis and programming. In C. Reading (Ed.), *Data and context in statistics education: Towards an evidence-based society. Proceedings of the Eighth International Conference on Teaching Statistics (ICOTS8, July, 2010), Ljubljana, Slovenia.* International Statistical Institute.

Wickham, H. (2017). Tidyverse: Easily install and load the 'tidyverse'. *CRAN.* https://CRAN.R-project.org/package=tidyverse

Wickham, H. (2018). *Should all statistics students be programmers?* Paper presented at the Tenth International Conference on Teaching Statistics (ICOTS10, July 2018), Kyoto, Japan. Speaker Deck:https://speakerdeck.com/hadley/should-all-statistics-students-be-programmers

Wiedemann, K., Chao, J., Galluzzo, B., & Simoneau, E. (2020). Mathematical modeling with R: Embedding computational thinking into high school math classes. *ACM Inroads*, *11*(1), 33–42. https://doi.org/10.1145/3380956

Wild, C. J. (2006). The concept of distribution. *Statistics Education Research Journal*, *5*(2), 10–26. https://doi.org/10.52041/serj.v5i2.497

Wild, C. J. (2015). Further, faster, wider. Online discussion of Cobb, G.W. (2015), "Mere renovation is too little too late: We need to rethink our undergraduate curriculum from the ground up". *The American Statistician*, *69*, 266–282. https://doi.org/10.1080/00031305.2015.1093029

Wild, C. J. (2018). Gaining iNZights from data. In M. A. Sorto, A. White, & L. Guyot (Eds.), *Looking back, looking forward. Proceedings of the Tenth International Conference on Teaching Statistics (ICOTS10, July, 2018), Kyoto, Japan.* International Statistical Institute.

Wild, C. J., & Halstead, B. (n.d.). VIT Online. https://www.stat.auckland.ac.nz/~wild/VITonline/

Wild, C. J., & Pfannkuch, M. (1999). Statistical thinking in empirical enquiry. *International Statistical Review*, *67*(3), 223–248. https://doi.org/10.1111/j.1751-5823.1999.tb00442.x

Wild, C. J., Elliott, T., & Sporle, A. (2021). On democratizing data science: Some iNZights into empowering the many. Commentary on Adhikari, A., DeNero, J., & Jordan, M. I. (2021), "Interleaving computational and inferential thinking: Data science for undergraduates at Berkeley". *Harvard Data Science Review*. https://doi.org/10.1162/99608f92.85206ff9

Wild, C. J., Pfannkuch, M., Regan, M., & Parsonage, R. (2017). Accessible conceptions of statistical inference: Pulling ourselves up by the bootstraps. *International Statistical Review*, *85*(1), 84–107. https://doi.org/10.1111/insr.12117

Wilkerson, M. H., & Laina, V. (2018). Middle school students' reasoning about data and context through storytelling with repurposed local data. *ZDM*, *50*(7), 1223–1235. https://doi.org/10.1007/s11858-018-0974-9

Wilkerson, M. H., & Polman, J. L. (2020). Situating data science: Exploring how relationships to data shape learning. *Journal of the Learning Sciences*, *29*(1), 1–10. https://doi.org/10.1080/10508406.2019.1705664

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

Wood, M. (2005). The role of simulation approaches in statistics. *Journal of Statistics Education*, *13*(3). https://doi.org/10.1080/10691898.2005.11910562

Woodard, V., & Lee, H. (2021). How students use statistical computing in problem solving. *Journal of Statistics and Data Science Education*, *29*(sup1), S145–S156. https://doi.org/10.1080/10691898.2020.1847007

Wouters, P., Paas, F., & van Merriënboer, J. J. (2008). How to optimize learning from animated models: A review of guidelines based on cognitive load. *Review of Educational Research*, *78*(3), 645–675. https://doi.org/10.3102/0034654308320320

Wright, S., Watson, J., & Fitzallen, N. (2019). Exploring one student's intuitive ideas about chance using TinkerPlots. *Australian Primary Mathematics Classroom*, *24*(1), 23–29.

Yang, Y., Liu, S., & Xie, N. (2019). Uncertainty and grey data analytics. *Marine Economics and Management*, *2*(2), 73–86. https://doi.org/10.1108/MAEM-08-2019-0006

Yoon, C., Patel, A., Radonich, P., & Sullivan, N. (2011). Learning environments with mathematical modeling activities. *Teacher and Learning Research Initiative.* New Zealand Council for Educational Research.

Zhang, I. Y., Tucker, M. C., & Stigler, J. W. (2021). Watching hands shuffle data improves subsequent understanding of R-based simulations of randomness. In R Helenius & E Falck (Eds.),*Statistics education in the era of data science. Proceedings of the satellite conference of the International Association for Statistical Education.* IASE.

Zieffler, A., & Catalysts for Change. (2019). *Statistical Thinking: A simulation approach to uncertainty* (4th ed.). Catalyst Press. http://zief0002.github.io/statistical-thinking/

Zieffler, A., Garfield, J., DelMas, R., & Reading, C. (2008). A framework to support research on informal inferential reasoning. *Statistics Education Research Journal*, *7*(2), 40–58. https://doi.org/10.52041/serj.v7i2.469

Zieffler, A., Justice, N., delMas, R., & Huberty, M. D. (2021). The use of algorithmic models to develop secondary teachers' understanding of the statistical modeling process. *Journal of Statistics and Data Science Education, 29*(1), 131–147. https://doi.org/10.1080/26939169.2021.1900759

Zohrabi, M. (2013). Mixed method research: Instruments, validity, reliability and reporting findings. *Theory and practice in language studies*, *3*(2), 254–262. https://doi.org/10.4304/tpls.3.2.254-262