Data Literacies in Informal Settings

Ruijia Cheng

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington
2023

Reading Committee:
Benjamin Mako Hill, Co-Chair
Jennifer A. Turns, Co-Chair
Sayamindu Dasgupta

Program Authorized to Offer Degree:
Human Centered Design and Engineering

University of Washington

Abstract

Data Literacies in Informal Settings

Ruijia Cheng

Co-chairs of the Supervisory Committee:

Benjamin Mako Hill
Department of Communication

Jennifer A. Turns
Department of Human Centered Design and Engineering

As data becomes an integral part of our lives, the general public faces the increasing need to actively engage with data to participate in daily activities, support personal goals, and understand social issues. Formal data science training, however, remains out of reach for most people and does not cater to their diverse needs related to data. Emerging informal settings, such as online and in-person social spaces and community workshops, offer accessible platforms for diverse and meaningful data engagements. However, current research on data literacy does not fully capture the diverse ways that the public interacts with data in these informal environments.

This dissertation presents four studies exploring the ways people interact with data in informal settings and examines the challenges and needs emerging from these engagements. These findings guide future research and shape the design of tools to foster data engagement in diverse informal environments. Study A illustrates a mixed-method analysis of 400 Scratch forum discussion threads and more than 240,000 user-generated projects, unpacking the benefits and drawbacks of interest-driven participation that involves data in a large online community. Study B presents a semi-structured interview study with 14 Kaggle users on their collaborative and communicative practices in working with large datasets, highlighting the needs and challenges in communicating procedures to a diverse audience and fostering collaboration among users of

different experience levels. Study C contains a theory-driven quantitative analysis of a large collection of Twitter messages that involve discussions about COVID-19 vaccine data, identifying features that differentiate critical engagement with data from conspiracy discourses. Study D presents a constructionist system that scaffolds novices to programmatically analyze and visualize data, as well as the insights from user study workshops that showcase the diverse range of concepts, perspectives, and practices that the system can support.

Together, these studies reveal a pluralism in people's competencies and epistemological pathways concerning data engagement—what I refer to as "data literacies"—that should be accounted for in the research and design of technologies for data literacies. This dissertation contributes rich empirical knowledge on the public's engagement with data in a range of informal settings, various design recommendations for informal environments to support data literacies, a call for acknowledging the pluralism in data literacies in the design of tools and interventions, and a sociotechnical framework for conceptualizing and designing to support data literacies in informal settings.

Acknowledgements

This dissertation could not have been made possible without the extensive support of many, many people. As I reflect on the journey to complete this dissertation, I am filled with gratitude for the invaluable help and guidance I received from various individuals and communities. I am deeply grateful for their involvement in my academic and personal growth during this challenging but extremely intellectually enriching period of my life. I therefore dedicate this section to express my appreciation to everyone who supported me through this journey. I try to acknowledge a lot of people in this section, but I am sure I still leave out some by accident. For those who I fail to mention in this section, this dissertation is my thank you to you.

First, I want to thank my committee. Mako, thanks for taking me in as a student even though I was not from your department, for always believing in me, and for being so available all the time and literally supporting every single step and building brick of my PhD journey—I hope that one day I could become a mentor like you. Thank you, Sayamindu, for the extensive mentorship, funding, and support, as well as the long, enlightening discussions about learning, data, and community—my work on data literacies could never have been carried out without your work, knowledge, and help. Jennifer, thank you for being my source of both intellectual and emotional support over the years and for always offering me a new perspective to look at my work and my journey. Thanks Amy for being my GSR and also providing me with invaluable feedback and suggestions.

I would like to thank all my coauthors and collaborators. None of my studies, included or not in this dissertation, would have been possible without your dedication and help. I learned a lot from EACH and EVERY one of you and I truly appreciate your effort and support. I also want to thank all of the students in the Directed Research Groups that I have participated in or led. All of your contributions have influenced, if not directly contributed to, the research in this dissertation.

I want to thank my main research group, the Community Data Science Collective. Quite literally, I randomly walked through the door of this group one afternoon in April 2019 and have never left since. I am lucky to have met the nicest, most helpful, and most creative people I know in this group. I have learned so much in every workshop, reading group, retreat, and every random conversation. Thanks Aaron Shaw for the constant support for my research and career development. Thanks Kaylea Champion, Emilia Gan, Stefania Druga, Nate TeBlunthuis, and Charlie Kiene for being such amazing and inspiring labmates, making my PhD journey enlightening and fun, and always answering my questions and offering me help. Thanks Jeremy Foote, Floor Fiers, Sohyeon Hwang, Carl Colglazier, Nick Vincent, and Molly de Blanc for always celebrating my success, encouraging me, and offering the most timely and insightful feedback I could ask for. Thanks everyone else in the group—I have grown so much in my research and as a person under the influence of everyone in CDSC.

I want to thank the HCDE and DUB community. Thanks Kathleen Rascon and Pat Reilly for being such excellent academic advisors and helping me navigate all the requirements of the PhD and Master programs and the complications of the CPT and OPT. Thanks Jane Skau for helping me figure out spaces and equipment for my studies and for organizing the HCDE Tea Times—always the highlight of a week! Allen Lee, Stacia Green, and Summer Dela Cruz Parkes, thanks for helping me figure out funding and my usually very late reimbursement requests. Thanks Yihan Yu for helping me survive the first year of grad school and for continually being a great friend. Ruotong Wang, thanks for the long and enlightening research discussions during and beyond our internships, and thanks for practicing my talks and dissertation defense with me. Thanks Jenna Frens for being the kindest mentor and the most supportive friend in my first year of grad school and beyond. Thanks Ray Hong for all the great research, career, and life advice. Thanks Anna Shang, Spencer Williams, Keri Mallari, Lotus Zhang, Himanshu Zade, Aayushi Dangol, and many others for being my grad school buddies and getting me through my PhD with all the support, rants, and inspirations.

I would like to thank my internship mentors. Although none of my internship projects are directly included in this dissertation, the internships I did have been some of the highlights of my grad school journey. I learned a lot from those experiences and was inspired by many people I met. Thanks Alison Smith-Renner and Ke Zhang, my mentors at Dataminr, for introducing me to the space of human-AI teaming and for being

such kind mentors to guide my exploration. Thanks Jonathan Grudin for guiding me in the Search Coach project at Microsoft. Thanks Denae Ford and Thomas Zimmermann, my mentors at Microsoft Research, for making my first in-person internship an unforgettable experience and for your great support in my projects and career development.

I want to thank the people at the UCSD Design Lab. As I spent a large chunk of my PhD in San Diego, I am fortunate to learn from and be friends with many of the professors, students, and researchers at the Design Lab. Thanks Steven Dow for hosting me as a visiting researcher in summer 2019, for supporting me in publishing my first research paper, and for helping me in all sorts of way from the very beginning of my research journey. Thanks Tone Xu for all the runs for brunch, dinner, dessert, and boba, and for all the impromptu research discussions. Thanks Stephen MacNeil and Brian McInnis for all the advice on my research and my career. Thanks all the friends at the Design Lab for sharing your space and resources and inviting me to your talks and parties.

I want to thank all my great friends outside of academia. As I spent most of my PhD journey working remotely, I truly appreciate that you all kept me company during the period of my life that would have been so isolated. Thanks Cindy Jia for being my best friend since college and for all the fun times. Sherry Ding, thanks for offering me your place to stay when I was in Seattle and visiting me in San Diego. Yao Wang, thank you for being such a great roommate, for your wonderful cat Koji, and for your heroic act of moving my furniture during the pandemic. Thanks Runjie Guan, for being a friend since high school and letting me have your cat Huli as an occasional company in my grad school days. Thanks Steve Zhao, for all the great restaurant trips and the deep life reflections that occurred during them.

Thank you, Yinan Xuan, my life partner, for your support in countless ways throughout this journey, each and every day. Thanks Hime, my cat, for being so perfect.

I want to thank my family for being so supportive and understanding of my dreams. Thanks my parents, Zhiqing Liu and Bing Cheng, for attending both my dissertation proposal and defense very late at night and for always trusting me and believing in me. Thanks my grandmother, Lijuan Han, for learning how to video chat with me at 95 years old and always supporting me. Thanks my grandfather, Shihua Cheng, for being the strongest supporter for me to pursue a PhD and a career in research. He unfortunately could not see this day since he passed away in 2020. To honor my grandfather, I dedicate this dissertation to him.

Last but not least, I want to extend my appreciation to all the participants of my study and to every member of the communities that I have studied. Thanks for your contributions, insights, and experiences that have been invaluable to my research in this dissertation.

Part of this dissertation is based on work supported by the National Science Foundation under Grant No. 2230291. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

DEDICATION

To all life-long learners and creators

DEDICATION

To Shihua Cheng

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Background

In today's world, there is little doubt that data, or collection of numbers and facts in other formats, has permeated every facet of our lives. Reflecting on my own experience as a graduate student during the remote work era of the COVID-19 pandemic: each morning, upon waking up, my smartphone and wearable device would present me with an array of data points, from my sleep patterns to physical activity levels, influencing my actions and decisions for the day ahead; my engagement with the broader world often involved "doom-scrolling" through social media and news platforms, where I attempted to make sense of the situation from an avalanche of data such as infection rates and vaccine distribution statistics; as I delved into my day's meetings and deep work, my devices would log productivity metrics and, if I like, could use the data to offer me personalized recommendations on my productivity. This reality stands in stark contrast to my childhood in the 2000s, when data was largely confined to be a specialized tool used by experts in scientific labs, financial offices, and other professional settings that were somewhat distant from our everyday lives. We have witnessed how far and how profoundly the landscape of data has evolved: It has become a common part of our lives, shaping how we understand, interact with, and even construct our world.

As data has become increasingly integrated and accessible in our lives, it is crucial to understand how the public actively engages with it. For example, the global pandemic of COVID-19 vividly illustrates the critical importance of public engagement with data. During the peak years of the pandemic, data on daily

infection rates, mortality statistics, hospital capacities, and vaccine distribution were broadcast on news and social media, reaching most people in the world. People would need to read, understand, and reason with such data to make decisions about daily activities like grocery shopping, use of public transport, and social gatherings.[1] Individuals' engagement with the data also fostered a sense of communal responsibility, as people collectively understood the gravity of their individual actions on community health. The collective interpretation of these data also shaped policies and official responses: the public would question whether governments, organizations, and community leaders had been making data-driven decisions, such as distributing vaccines and reopening businesses and schools.[2] These examples are just snapshots of how prevalent and crucial it is for everyone to engage with data.

When thinking of promoting data engagement in the public, many would point to formal education programs and curricula that focus on programming and statistical education. In fact, higher education programs in computer science and data science have seen a surge in enrollment in recent years (West and Portenoy, 2016). Massive Open Online Courses (MOOCs) in the same fields have proliferated on platforms such as Coursera and Udemy. Even at the K-12 level, data science curricula have emerged and received increasing attention (Lee et al., 2022). However, formal education in these areas faces several challenges. Despite the increasing numbers of programs, data science education remains unreachable for many, and certain populations find it particularly inaccessible. For example, in 2022, the University of Washington School of Computer Science received a record number of more than 8,500 applications, while only 26% of applicants from Washington state and a starkly low rate of 2% from outside the state were accepted.[3] As for MOOCs, despite the intention to expand computing education, only 13% of the students would be able to complete the courses they start with, and women in particular faced greater systematic barriers to completing MOOC courses (Dai et al., 2022).

At the same time, the public's growing need to engage with data does not necessarily translate into the desire to become professional data scientists. Many want to harness the power of data to answer questions, make decisions, or solve problems in their daily lives without committing to the formal education path

[1] `https://www.apa.org/news/press/releases/stress/2021/october-decision-making` (permalink: `https://perma.cc/J8K6-S49Y`)

[2] `https://blogs.worldbank.org/governance/how-governments-can-use-data-fight-pandemic-and-accompanying-infodemic` (permalink: `https://perma.cc/G297-TZJY`)

[3] `https://www.cs.washington.edu/academics/ugrad/admissions/direct` (permalink: `https://perma.cc/7Y3X-KHCQ`)

and the specialized career. Indeed, for the vast majority, opportunities to learn about and with data often occur organically in a variety of informal settings, both online and offline, including community workshops (e.g., (Hill et al., 2017)), and a variety of online platforms where people create, share, and interact around artifacts and discussions involving data (e.g., (Kauer et al., 2021)). Understanding these informal settings thus becomes crucial, as they hold great promise for the public to engage with data, potentially offering valuable insights for the development of strategies, tools, and learning environments that are both accessible and resonant with a diverse range of learners.

1.1.1 Data Literacy

To facilitate effective public engagement with data, the research community has been striving to *data literacy* among the general public. Usually referring to a range of competencies, skills, concepts, practices, and perspectives around data, at present, researchers predominantly pay attention to two main bodies of data literacy, described as follows:

The first body emphasizes the promotion of statistical and computational skills related to data. This includes a range of competencies such as the ability to read data (e.g., understanding what a dataset represents), work with data (e.g., cleaning a dataset), analyze data (e.g., filtering data), and argue with data (e.g., using data to support a specific plan of action or message) (D'Ignazio and Bhargava, 2015). Many of the tasks of reading, working with, analyzing, and arguing with data typically require the *technical* ability to write computer programs (Berman et al., 2018; CrowdFlower, 2017). Numerous sources have identified such skills as essential abilities to meaningfully work with data. For example, Prado and Marzal (2013) listed the ability to "handle and analyze" data as a key competency for students to master in a course designed to promote data literacy. Deahl (2014) described "numerical and quantitative literacy" in data science learning as the equipment of relevant mathematical and statistical knowledge and computational thinking skills. D'Ignazio and Bhargava (2015) pointed out that, most of the stages of working with data require programming or other technical skills. For instance, when developing an analysis plan, learners sometimes need to make a new variable to store an exploratory measurement; when making visualizations to present their findings, learners need to have statistical knowledge to construct informative figures.

The second body has paid attention to the abilities to realize the community and social impact of data

and has named them *critical data literacy*. In recent years, discussions considering learners' critical understanding of where data come from and how data are used have been emerging. In previous research, the term "critical data literacy" has been used to describe the ability to "access, interpret, critically assess, manage, handle, and ethically use data" when considering the larger social context (of Data Institute staff, 2016). Deahl (2014) argued that, beyond quantitative measures, learners should be able to collect and pay attention to "qualitative data", which includes background information on data collection and its impact in a broader social context. Similarly, D'Ignazio (2017) encouraged data science learners to write "data biographies" to acknowledge how the data came into the world and recognize the invisible works and underlying assumptions. In another work on "big data literacy", D'Ignazio and Bhargava (2015) argued for the importance for the general public to identify when and where their data are collected, understand the algorithmic manipulations behind everyday technology, and weigh the individual and community impacts of data-driven decisions. Hautea et al. (2017a) summarized the emerging scholarly consensus on what constitutes critical data literacy as the ability to understand "the implications of large-scale data collection and analysis" and critically think about "the issues of privacy, surveillance, and power structures that enable data-driven processes to affect people's lives." All these discussions agree that data science learners need to develop the ability to interpret and argue for the impact of data in a larger social context.

While these are established understandings of data literacy, it is debatable whether these two predominant perspectives are sufficient to help us understand the public's engagement with data in informal settings. On the one hand, many of these studies on data literacy stem from structured environments such as classrooms and professional settings, leaving a gap in the examination of data literacy "in the wild." Informal spaces, on the other hand, present a richer setting, distinguished by diverse user groups, various forms of engagement, and a mix of platform affordances and underlying social structures. For example, in a wide variety of informal settings—from online social platforms and competitive forums to community workshops—we witness participants whose backgrounds and motivations span a spectrum: from those who informally engage with in interest-driven media creation, to proactive learners diving into data science; from the curious minds seeking personal answers, to citizens using data to support civic discussions.

1.2 Research Questions

Given such diversity, little is known about the ways these diverse groups engage with data in such different and organic settings and the challenges in their way. This dissertation aims to explore the diverse ways in which the public engages with data, providing insights to inform future research and technology design to promote data literacy in informal settings. It centers on these overarching research questions:

RQ1: How do people engage with data in informal settings, and what challenges and needs arise in these engagements?

RQ2: What can we learn from these informal engagements to foster public engagement with data?

1.3 Dissertation Contribution

The dissertation presents four studies that investigated these research questions across a variety of informal settings, featuring a diverse range of learners. These studies reveal a pluralism in people's practices, competencies, and needs concerning data engagement, or what I refer to as "data literacies" in their plural form, that can be supported in informal settings. The foundation for supporting these diverse forms of data literacy lies in the sociotechnical nature of data engagement. Specifically, the ways in which people interact with and understand data are influenced by the particular affordances and communities of practice inherent to these informal settings.

This dissertation presents the following thesis statement:

> The public's engagement with data in informal settings presents a pluralism in both the different competencies that people should seek and the diverse epistemological pathways to obtain the competencies. The design and research for data literacies in informal settings should be situated in the affordances and context of the given informal setting to support the domain, community, and practice.

Specifically, this dissertation makes the following contributions:

- This dissertation contributes rich empirical knowledge on the public's engagement with data in a range of informal settings. The four empirical studies presented in this dissertation illustrate a set of distinct competencies and pathways regarding data literacies that extend beyond merely programming, statistical skills, or critical data engagement.

- For each of the four studies, this dissertation contributes a range of design considerations for informal environments that support various data literacies.

- This dissertation calls for the design of tools and interventions for data literacies to acknowledge the pluralism that accounts for the different competencies that people should seek when engaging with data across various contexts and communities, as well as the diverse epistemological pathways to get to the competencies.

- In the end, this dissertation makes a theoretical contribution by offering a new sociotechnical framework for conceptualizing and designing to support data literacies in informal settings.

1.4 Overview of the Studies and Methods:

In particular, my studies serve as concrete illustrations of the varied ways in which the public engages with data, as well as the unique challenges and opportunities each setting presents. Specifically, I conducted empirical studies on a range of public on-line platforms: Scratch (Study A), Kaggle (Study B), and Twitter (Study C), as well as in offline workshops centered on a research prototype, *Dataland* (Study D). While it is impractical to examine every type of informal setting where engagement with data occurs, the settings chosen for these studies serve as representative examples, featuring key aspects of different types of data engagement, user communities, platform infrastructure, and social dynamics.

These studies spanned from 2019 to 2022 and are roughly chronologically organized in this dissertation. However, the order of the presentation is not particularly crucial, as each study stands independently, representing distinct facets of the overarching thesis.

As an overview, Study A focuses on the experiences of novices who engage with data in a playful, social, and participatory media creation online community. In particular, I discovered community dynamics that shape the learning of programming concepts related to data and the benefits and drawbacks.

Study B focuses on adults actively seeking to improve their data science skills by participating in open data science competitions. In this study, I explored the practices and challenges faced by beginners and experts in collaborating and communicating procedures of working with data. I revealed how these interactions are influenced by the community's unique blend of competitive and collaborative social structures.

Study C looks at social networks where the public actively participates in discussions about data as a form of informal civic engagement. This exploration centers on the risk of critical engagement with data

falling for conspiracy claims and an effort to computationally distinguish the two. This study presents implications for online platforms to foster critical civic engagement with data on public issues while addressing the risks of conspiracy discourses.

Study D demonstrates an investigation on "Dataland", a constructionist platform that I participated in building to scaffold middle and high school students to programming with data in the context of data-driven investigation workshops. This study reveals the concepts, practices, and perspectives of computational data literacy that students develop with the support of the system.

Due to the diverse settings in these studies, I used a range of empirical methods for data collection and analysis. In Study A, I conducted a grounded theory analysis of 400 online discussion threads, developed a theoretical model, and performed quantitative hypothesis testing of the model using a large-scale collection of online user activity logs. For Study B, I conducted semi-structured ethnographic in-depth interviews. In Study C, I engaged in a quantitative analysis of large volumes of social media messages. Lastly, Study D involved the design and implementation of a research system, as well as a series of research workshops where users tried the system out.

1.5 Dissertation Structure:

This dissertation is structured as follows:

Chapter 2 lays the theoretical groundwork for the studies in this dissertation,particularly focusing on the theory of communities of practice and its application in social computing research; the learning theory of constructionism, and its application of computational participation.

Chapters 3 through 6 present a series of four studies, along with the description of their methods, empirical settings, findings, as well as discussions and implications for technology design.

Chapter 3 presents Study A, containing materials from a paper originally published in CHI'22.

Chapter 4 presents Study B, containing materials from a paper originally published in CSCW'20.

Chapter 5 presents Study C, containing materials from a manuscript that discusses social media conversations about COVID-19 data and public critical engagement,

Chapter 6 presents Study D, containing materials from a paper originally published in IDC'23.

Chapter 7 synthesizes these studies to draw insights on the pluralism in data literacies in informal settings

and the implications for design, leading to a new sociotechnical framework for studying and designing for data literacies. It concludes with a look at the future of studying and supporting data literacies given the emerging generative AI technology.

Finally, Chapter 8 concludes the contributions of the dissertation.

Chapter 2

Theoretical Frameworks

This chapter outlines the general theoretical foundations underpinning this dissertation. Individual studies may further elaborate on specific theories in their corresponding background sections.

2.1 Online Communities of Practice

The first thread of literature that influences my work is about online communities of practice and its role in learning. As the development of data literacies involves advancement of many different skills, it requires a learning context that supports plural learning goals and pathways. Drawing from previous research on online communities of practice, I argue for the potential of online communities to support various kinds of learning. The following sections introduce the theory of community of practice and relevant examples in the social computing literature. This section presents unmodified text from the background section of a paper that I published at the ACM SIGCHI Conference on Computer-Supported Cooperative Work & Social Computing in 2022 (Cheng and Hill, 2022). The full paper is attached in the Appendix C.

2.1.1 Communities of practice

Over the last three decades, Jean Lave and Etienne Wenger's ideas around CoP have been among the most important theories used by social computing scholars to understand how learning happens in communities. The term *community of practice* has been widely used in social computing scholarship to describe groups of people learning from each other while working toward a common interest or goal (Shrestha et al., 2021; Mar-

low and Dabbish, 2014a; Kou et al., 2018; Gilbert, 2016). First introduced by Lave et al. (1991), CoP theory is grounded in ethnographic observation of apprenticeship relationships in communities of Liberian tailors, Mayan midwives, US Navy quartermasters, nondrinking alcoholics, and US supermarket meat cutters.

Learning in CoPs is described as occurring through LPP, the phenomenon through which newcomers begin to participate in a group by helping out with tasks that are easy and low-risk but still valuable and important. For example, a new midwife may begin by boiling water and cleaning scissors for other midwives. Through observing more experienced members performing more complex and higher-stake tasks—and by practicing themselves in various ways—novices move from the periphery to more central roles.

2.1.2 Applications of CoP theory in social computing research

Although created to explain learning through traditional apprenticeships offline, the HCI and social computing communities have embraced the CoP framework. Today, it is one of the most influential and highly cited theories related to learning in social computing. CoP has been widely adopted by social computing scholars because it is a good match for informal and individualized forms of learning that occur in many online settings.

Furthermore, the work of applying CoPs to online settings, where groups are often extremely porous and diffuse, has required new theoretical work and argumentation. For example, Gruzd et al. (2011) have argued that a single social media user can form a personal or "imaginary" CoP with other users in their network. Using a similar line of thinking, data scientists on Twitter who engage with the hashtag "#TidyTuesday" have been theorized to constitute a CoP because they share context, common interests, and a process of collaborative knowledge advancement and because they ask questions and interact with more experienced users (Shrestha et al., 2021).

A wide range of other examples of online CoPs that have been identified and studied in social computing include design professionals on online critique platforms (Marlow and Dabbish, 2014a; Kou et al., 2018), a Facebook group of Airbnb hosts who learn about new features and hosting strategies from each other (Holikatti et al., 2019), and fan fiction authors who gather online to develop and maintain the fan fiction website Archive of Our Own (AO3) (Fiesler et al., 2017a). In these examples and others, online CoPs are generally described as occurring in *affinity spaces* where members with similar interests and identities

contribute to a shared collection of knowledge in a distributed manner (Gee, 2005). An example of an online affinity space that is frequently described as a CoP is the Scratch online community.

2.1.3 Types of learning in CoPs

One limitation in Lave and Wenger's initial account is that it is vague about what exactly is being learned in a given CoP. This is an important omission because CoP members typically learn a range of different things as they become more experienced. For example, becoming a Mayan midwife in the Yucatan Peninsula involves much more than learning technical midwifery skills. It also involves learning about the Mayan midwifery community and the specific norms and values that shape midwifery in the Yucatan. In later work, Wenger et al. (2002) attempted to address this omission by identifying three types of learning in CoPs: learning about *domain*, *community*, and *practice*.

First, learning about a *domain* refers to the acquisition of knowledge and skills necessary for a person to carry out the core tasks at the heart of a CoP. Many scholars of online communities are interested in how communities use LPP to support learning about some domain of knowledge and domain learning is often assumed to be the only learning goal in a CoP. For midwives, learning about a domain involves gaining knowledge related to successfully delivering infants. Depending on the community, domain knowledge may involve skills related to computer programming (Resnick et al., 2009b; von Krogha et al., 2003), fan fiction writing (Campbell et al., 2016a), encyclopedia article editing (Halfaker et al., 2013b), and so on. In the specific context of computational learning communities such as Scratch, domain learning means learning computational concepts and related programming skills (Resnick et al., 2009b).

The second type of learning involves the development of identity as a member of the *community*. This involves developing relationships, affinities, and a sense of belonging. As learners are accepted by older members of the community, they gradually form an identity as a member of the community. The development of these relationships and affinities would play out similarly in various settings and typically involves the knowledge of other community members and the development of a sense of membership and commitment to the community (McMillan and Chavis, 1986).

Third and finally, learning a *practice* means assimilating "cultural artifacts, norms, and values" developed in the community over time (Barab and Duffy, 2000). By moving from peripheral to central forms

of participation, learners develop by adjusting their social, work, and contribution style to match what is accepted and appreciated by community members. The specific reasons why one would be seen as accepted and appreciated in a CoP are very community-specific. However, indicators of practice development will typically involve expressions of appreciation and respect from others. For example, for midwives, it might involve the authority to lead or manage other midwives (Lave et al., 1991). On Fanfiction.net, signs of practice development may involve high ratings and positive comments (Campbell et al., 2016a). On Scratch, it might involve "loves" (i.e., likes) given by other users (Brennan and Resnick, 2013).

2.2 Constructionism

2.2.1 The constructionist learning theory

Another line of theory that is fundamental to this dissertation is constructionism, a learning theory first proposed by Papert (1993). Constructionism stems from Piaget's constructivism, which recognizes learners as active constructors and discoverers of knowledge rather than passive recipients of knowledge transmitted by others (Tsou, 2006). Building on this line of thinking, Papert emphasized the importance of concrete entities in this knowledge construction process. He believed that learning would happen "especially felicitously" when learners actively create and interact with concrete entities, such as building blocks and computer programs (Papert, 1991a). Constructionism also emphasizes the importance of "personal relevance" in learning, that is, the features of a learning system should be continuous with the personal knowledge of the learners and allow the learner to perform personally meaningful projects (Papert, 1993). With such "objects to think with", learners could externalize their ideas and thinking, feeling personally interested and empowered. Constructionism has informed the design of many learning systems. I have synthesized three major characteristics of constructionist designs as follows:

Concrete epistemological relevance: Constructionist tools should provide ways to connect the concrete object and material that the learner would use to make projects to the domain and skills the learner is trying to learn (Resnick et al., 2005). Learning systems should strive to "create frameworks from which strong connections and rich learning experiences are likely to emerge" (Resnick et al., 1996a). For example, as a successful constructionist system for computational learning, Scratch offers learners visual blocks that were

thoughtfully designed based on computational concepts. Learners could control the behavior of graphical objects on the screen by dragging and dropping blocks together. In this process, learners are able to play with concrete examples, see tangible effects, and learn about otherwise abstract programming logic and concepts.

Personal and community relevance: Constructionist learning systems should also allow learners to make projects that they like to make and are relevant to their interests. In his example of "appropriate math", Papert stated that the features of the system should be continuous with the personal knowledge of the learners and could enable the learner to perform personally meaningful projects (Papert, 1993). Later scholars added that the system should support activities that make sense in terms of a larger social context. Good constructionist tools can support the formation and prosperity of communities of practice, in which learners could work toward projects that are valuable and meaningful to the community (Bruckman, 1998).

Epistemological Pluralism: Constructionist learning systems should support "epistemological pluralism" — different pathways to gain knowledge (Papert, 1991a). Learners should have space for explorations and alternative learning pathways that support the different needs of learners with a diverse background and experience. For example, constructionist tools should provide learners with "low floors" — low entry bar for beginners, "high ceilings" — the opportunity to make sophisticated projects, and "wide walls" — playground to explore different approaches to build an artifact with low cost of trial and error (Resnick et al., 2005).

2.2.2 Computational Participation

The related line of work that inspires this dissertation is "computational participation". Computational participation stems from Kafai (2016)'s call for learners to shift from creating programs for structured coursework toward creating programs to be shared with peers in interest-driven and socially supported contexts. Although primarily focusing on programming, the model of computational participation provides inspiration for promoting data literacies outside formal education programs in informal, interest-driven, and accessible ways.

Computational participation has its root in *constructionism*. Later, constructionist scholars added that

learning systems should support activities that make sense in terms of a larger social context, empowering learners to work toward projects that are valuable and meaningful to their community (Bruckman, 1998). Recent scholarship on connected learning (Ito et al., 2013, 2018) has also endorsed the role of shared interest and participatory culture in building learning communities.

In recent years, online communities have emerged to be a prominent context for computational participation. Some notable examples of these communities include programmable multiplayer game environments (Bruckman, 1998), platforms for interactive media creation (Monroy-Hernández, 2007; Resnick et al., 2009a; Wolber et al., 2011), and amateur technical support groups (Fiesler et al., 2017b). In all these examples, online communities support computational participation by following the two major recommendations of constructionism: offering members the opportunity to create "object to think with" and empowering them to engage in personal and community-relevant creation. Specifically, these learning opportunities take place through the pathways of sharing interactive computer programs (Dasgupta, 2016) and social interactions around these artifacts, such as commenting, remixing, and critiquing. To ensure learning with "object to think with", many communities are structured so that the learning products are made visible as public artifacts that others can use as illustrative examples and for inspiration (Dasgupta et al., 2016b; Gan et al., 2018) as well as scaffolds for replication, practice, and innovation (Dasgupta et al., 2016b; Tausczik and Wang, 2017a). To empower personal and community-relevant learning, communities support interest-driven socialization and direct user-to-user interaction such as comments, forum posts, and Q&A discussions (Tausczik et al., 2014a; Shorey et al., 2020a). In particular, computational participation allows indirect and direct input from experts and professionals that is otherwise unavailable. For example, by making artifacts publicly visible, learners can receive help and constructive feedback from online strangers (Gan et al., 2018). They can also participate in collaborative learning activities such as debugging (Shorey et al., 2020a) and sharing the repository of examples (Tausczik and Wang, 2017a).

As summarized above, the literature on community of practice and computational participation indicates the potential of informal settings such as online social spaces as a promising context to promote data literacies. However, it remains unknown how features and social dynamics in online communities shape the advancement of data literacies. My dissertation aims to build on these strands of work to uncover such dynamics.

Chapter 3

Study A: Scratch Online Community

Prologue

I conducted Study A in 2019 and 2020, focusing on the Scratch Online Community. My motivation was to understand how novices understand programming concepts related to data through computational participation in informal environments. Scratch Online Community is an example of such environments as it serves as a platform where novice programmers (mostly children) interact with computational concepts in a playful, social, and collaborative way and sharing and interacting around their creations. Notably, Scratch explicitly declares programming concepts related to data within its block-based programming scaffolds, making it convenient to study user interactions with and around those concepts by analyzing user-generated code corpuses. In particular, users can incorporate data into their programming projects using the basic data structures of scalar variables and lists, both of which have been recognized as challenging for beginners. I sought to investigate how novices understand these data structures in such an informal learning environment, how social dynamics in the community shape their understandings, and what generalizeable insights could be drawn to facilitate learning of programming concepts related to data in similar informal social environments.

Specifically, I performed a mixed-method analysis of user-generated tutorials around variables and lists. I first developed a social feedback loop theory from a grounded theory analysis on 400 Scratch discussion threads: specific user interests that frame the tutorials can be amplified over time, biasing the collective

understanding of what data is and how data can be used, and as a result, narrowing long-term innovation in the community. I then conducted a series of quantitative analyses on more than 240,000 user-made projects and found evidence in support of this theory—for example, I found that users who were exposed to popular use cases of data structures tended to make the same thing in their own projects. I list a series of design implications for community-based learning systems to support the diverse interests of learners.

The subsequent sections of §3.1 to §3.8 in this chapter feature the paper I published at the ACM Conference on Human Factors in Computing Systems in 2022 (CHI'22) (Cheng et al., 2022a). The paper was a collaborative effort between myself, Sayamindu Dasgupta, and Benjamin Mako Hill, in which I was the lead of the work and was responsible for the preparation of the datasets, the design and conduct of the analyses, and the writing of the manuscript. The content of the paper remains in its original form without modifications. §3.9 provides a short summary of the takeaways of this study with respect to the general thesis of this dissertation.

3.1 Introduction

Scholars increasingly look to interest-driven online communities as promising environments for supporting learning (Bruckman, 1998; Ito, 2009; Jenkins et al., 2009). These communities rely on user engagement in content creation to curate community-produced learning resources where users engage in sharing artifacts that they create and online discussion with other users. Although such communities exist in a range of domains like creative writing (Campbell et al., 2016b), graphical design (Marlow and Dabbish, 2014b), and more, many of the largest and most studied have focused on supporting young people in computational learning (i.e., learning about computational concepts, often through learning to program a computer). Widely studied examples include the Scratch community (Monroy-Hernández, 2007) and MOOSE Crossing (Resnick et al., 1998). These interest-driven computational learning communities are built around the idea of "computational participation" (Kafai and Burke, 2014), which encourages learners to develop programming skills through creating and sharing projects and interacting with other learners (Kafai, 2016; Bruckman, 1998; Brennan et al., 2011).

Despite this promise, it remains unclear how the creation and usage of community-generated learning resources support the learning of computational concepts. Previous studies of the Scratch online commu-

nity show that users who remix, or build projects from code shared by other users, achieve better learning outcomes (Dasgupta et al., 2016b). However, most users do not demonstrate much innovation in remixed projects, making people question whether they actually develop transformative abilities (Hill and Monroy-Hernández, 2013). In fact, most Scratch users only display a limited range of programming skills (Matias et al., 2016; Yang et al., 2015a). For example, only around 15% users have ever used data structures—an important computational concept—in Scratch projects (Dasgupta et al., 2016b). While online discussion provides opportunities for learners to mentor each other and collaboratively debug programs (Fields et al., 2015; Shorey et al., 2020a), it can also end up as superficial socialization in ways that can even act as a barrier to the exchanges of ideas, feedback, and resources (Shorey et al., 2020a). These mixed signals suggest the need of a better understanding about the mechanism of interest-driven content creation in computational learning communities. How does computational learning happen through interest-driven content creation? What is the role of community in the process? How do community-produced learning resources support learners' diverse interests?

To explore these questions, we present two studies about the Scratch community that describe the opportunities and challenges that interest-driven content creation and related community activities introduce to computational learning. In Study 1, where we present a grounded theory analysis of 400 discussion threads in the Scratch forums about how learners develop an understanding of data structures—variables and lists. Based on this analysis, we hypothesize a social feedback loop where engagement in content sharing and Q&A naturally raises the visibility of some particular ways of using variables and lists. Through their increased visibility, these examples become archetypes that can limit the breadth of future projects in the community. In Study 2, we conduct a quantitative analysis of the code corpus of more than 200,000 Scratch projects to test our hypothesis and find statistical support for the social process theorized in our first study. We conclude with several implications for design and content curation that we believe could improve support for diverse interests in Scratch and similar interest-driven learning communities.

This work makes several contributions to the HCI and social computing literature on computational learning. First, we make an empirical contribution by presenting detailed qualitative and quantitative evidence about how novices learn to use data structures in the Scratch online community. Second, we offer a theoretical contribution in the form of a framework describing a dynamic process where interest-driven

content creation can both assist learning about particular topics while posing important limits on the ways that those topics are engaged with. Finally, we make a contribution to the literature on the design of informal learning systems by speculating about how online interest-driven communities can be designed to mitigate the negative repercussions of the dynamic we describe.

3.2 Background

Online communities are increasingly common settings for participatory, interest-driven, and community-supported learning (Bruckman, 1998; Gee, 2006; Jenkins et al., 2009, 2016). A broad range of theoretical frameworks have been used to design and analyze these communities—many building on foundational theories on the social origins of learning by Vygotsky (1978) and Lave and Wenger (1991). Another key theoretical framing is Papert (1991b)'s view of learning as the construction of knowledge that "happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity" (p. 1) and which emphasizes the importance of interest-driven exploration and "personally powerful ideas" in promoting learning (Papert, 1993). Recent scholarship on connected learning (Ito et al., 2013, 2018) have also endorsed the role of shared interest and participatory culture in building learning communities.

Learning experiences in interest-driven online communities happen through two primary pathways: through sharing creative artifacts like fan fiction (Campbell et al., 2016b), design mock-ups (Cheng et al., 2020), interactive computer programs (Dasgupta, 2016); and through social interactions around these artifacts like commenting, remixing, and critiquing. To promote the first pathway, many communities are structured so that the products of learning are made visible as public artifacts that can be used by others as illustrative examples and for inspiration (Dasgupta et al., 2016b; Gan et al., 2018; Marlow and Dabbish, 2014b) as well as scaffolds for replication, practice, and innovation (Dasgupta et al., 2016b; Tausczik and Wang, 2017a). To promote the second pathway, communities feature direct user-to-user support such as comments (Campbell et al., 2016b), forum posts (Kou and Gray, 2017a), and Q&A discussions (Tausczik et al., 2014a) that can help members gain an understanding of specific topics or techniques (Shorey et al., 2020a). The two pathways are deeply interwoven. By making artifacts publicly visible, creators are able to receive constructive feedback that can support learning (Gan et al., 2018; Yen et al., 2016a). This often includes input from experts and professionals that is otherwise unavailable (Kou and Gray, 2018; Hui et al.,

2019) as well as social recognition and support (Campbell et al., 2016b). Additionally, learners in online communities often center their social interaction around discussions of public artifacts and as social interactions support the further production of artifacts (Kim et al., 2017a), collaborative problem-solving (Tausczik et al., 2014a; Li et al., 2015a; Shorey et al., 2020a) and community-wise knowledge advancement (Gray and Kou, 2019).

In recent years, creative programming communities have emerged as a prominent example of online interest-driven learning communities. In what Kafai (2016) has described as "the social turn in K–12 computing" (p. 27) and "computational participation," scholars have turned to interest-driven and socially supported contexts to promote learning about computing where learners create programs to be shared with peers. Through the work of efforts inspired by this approach, millions of young people have engaged in programming in online communities. Some notable examples of these communities include programmable multiplayer game environments (Bruckman, 1998), platforms for interactive media creation (Monroy-Hernández, 2007; Resnick et al., 2009a; Wolber et al., 2011), and amateur technical support groups (Fiesler et al., 2017b).

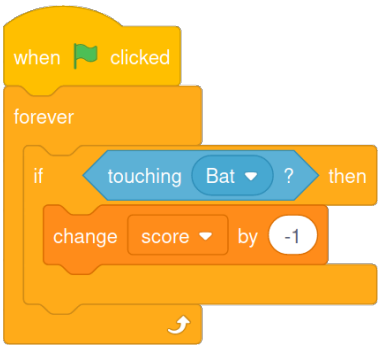
Despite the promise of these communities, it remains unclear how to best promote computational participation. Recent studies of online programming communities have shown unequal outcomes in terms of both participation and learning based on gender and race (Fields et al., 2014; Gan et al., 2018; Richard and Kafai, 2016) and that important debugging or collaborative sense-making activities are not always helped by socialization (Shorey et al., 2020a). Furthermore, while online communities allow users to gain inspiration from examples posted by others, studies on remixing activities indicate it can negatively impact originality (Hill and Monroy-Hernández, 2013). These examples indicate a lack of a general understanding of the dynamics around learning in online informal contexts. In interest-driven communities of all types, learning pathways can be blocked by the difficulty of ensuring high quality content (Kotturi and Kingston, 2019; Marlow and Dabbish, 2014b; Hui et al., 2019; Xu and Bailey, 2012a; Agichtein et al., 2008a) and the challenge of engaging diverse groups of users (Cheng and Zachry, 2020a; Buechley and Hill, 2010). This mismatch between the theoretical promise of online interest-driven communities and what is seen in practice indicates a lack of understanding of why user engagement supports or fails to support learning and how we can best design to facilitate positive learning outcomes.

Because computational participation involves learning many different concepts, we focus on learning experiences over one specific computational concept that has been the subject of substantial academic work: the simplest data structures comprising scalar variables and lists. We consider data structures specifically because Brennan and Resnick (2012a) identify the ability to understand how to store, retrieve, and update data as one of seven major practices that contribute to computational thinking (Barr et al., 2011; Denning and Tedre, 2019; Wing, 2006). Despite their importance, research has shown that data is the least commonly engaged computational thinking concept in Scratch (Dasgupta et al., 2016b). Previous work by Dasgupta et al. (2016b) estimates that less than 15% of Scratch users will ever make projects using data structures. When used, it is often engaged with in superficial ways (Blikstein, 2018).

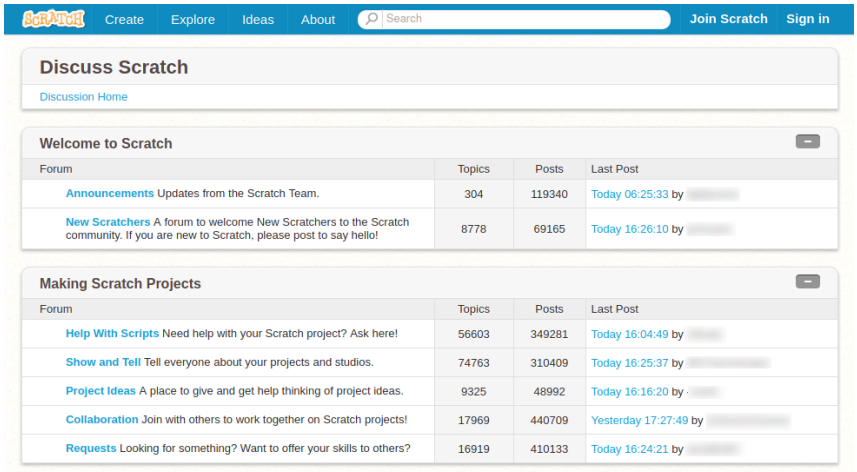
Why are data structures hard to learn? An explanation stems from the fact that learning about a computational concept involves learning its structural and functional uses (diSessa, 1986). The *structural uses* of variables and lists (i.e., how to integrate them in a program) are straightforward. For example in Scratch, there are only two methods (`get()` and `set()`) that represent the structural usage of scalar variables. However, the *functional uses* of variables and lists (i.e., what meaningful outcome that they can help create) are both broad and invisible. For example, while variables can be used for storing user input, keeping track of internal program state, counting in a loop, and so on, these functions may not be immediate obvious to novices. Previous work has argued that learners require "specific tutoring" (diSessa, 1986, p. 207) and has demonstrated that scaffolds are required to cope with misconceptions about how variables can be used to achieve concrete functionality (Hermans et al., 2018). It is unclear where interest-driven communities like Scratch are effective in providing proper tutoring needed for learning the functional uses of data structures. Therefore, we ask the following research question: *When does interest-driven content creation most effectively support learning of functional uses of variables and lists? When does it fall short?* We seek to answer these questions through two closely linked empirical studies that unpack practices, challenges, and opportunities for learning about variables and lists in the Scratch online community.

3.3 Empirical Setting

Scratch is a visual, block-based programming language designed for children (Resnick et al., 2009a). Scratch programming primitives are represented by visual blocks that control the behavior of on-screen



(a) Sample Scratch code showing a variable in the form of a data block called "score" being decremented on collision with a bat sprite



(b) Index page of the Scratch forums

Figure 3.1: The Scratch programming language and online community

graphical objects called sprites. Scratch programs (commonly called projects) are constructed by dragging and dropping blocks together. As a programming language, Scratch supports basic data structures in the form of scalar variables and vector lists (Figure 3.1a).

Primitives to operate on variables and lists fall under the category of "data blocks" in Scratch and their design are described in detail by Maloney et al. (2010). When creating a variable or list, Scratch users assign a name to the variable or list through a free-form text entry field that is invoked through a "make a variable" or a "make a list" button in the Scratch user interface. We refer to these user-defined names as "variable name" or "list name" in this chapter. When referring to the the variable or list within the Scratch program, users select the appropriate name from a dropdown list embedded in the block. Variables in Scratch have two forms which share a grammar: (1) conventional variables and lists which are local to each instance of a running project; and (2) cloud variables which are persistent across multiple executions of a project and shared across users (Dasgupta, 2013). Cloud variables are only accessible to established members in the Scratch online community as established by an undocumented set of criteria (Dasgupta and Hill, 2018a).

Scratch is situated within an online community where anyone can sign up and share their projects, comment on, "like," and bookmark others' works, and socialize in forums (Monroy-Hernández, 2007).[1] As of 2021, the Scratch online community has over 65 million registered users, and over 68 million shared projects that span a diverse range of genres and themes. The large majority of Scratch users are between 8–

[1] `https://scratch.mit.edu/` (permalink: `https://perma.cc/BRQ9-M3D6`)

16 years old and the median age for new contributors is around 12.[2] Although our data might include adults, we follow other scholarly accounts and refer to Scratch users as "kids" (e.g., Ito, 2009). We draw data from both the Scratch community itself and its discussion forums, shown in Figure 3.1b. These forums comprise a number of topical forums organized into categories such as "Making Scratch Projects" and subcategories such as "Help With Scripts" or "Project Ideas" (Scaffidi et al., 2012).[3]

3.4 Research Ethics

This research relies on two sources of data and included no intervention or interaction with subjects. Although the data in our qualitative analysis (§3.5) are public posts in the Scratch forums, we sampled from these posts using keyword searches of a copy of the Scratch Forums database in a way that the public could not do easily. We have obscured users' identities by replacing usernames with alphanumeric identities and by following advice from Markham (2012) to reword quotes to make it more difficult to identify the Scratch users we quote using simple web searches. Our quantitative analysis of the Scratch code corpus in §3.6 relied on data that the Scratch team has published as part of the *Scratch Research Dataset* (Hill and Monroy-Hernández, 2017). This work was reviewed and overseen by the IRB at MIT as part of a broader protocol covering observational studies of Scratch. Our institutional IRBs delegated oversight of the work on this project to the MIT IRB.

3.5 Study 1: Theory development

Because we could not identify existing theories with clear prediction on when Scratch would best support learning about variables and lists, we began with an open-ended interpretive analysis that sought to build a theory about kids' practices, learning, and engagement in discussions about simple data structures.

3.5.1 Methods

[2] All statistics about Scratch community activity and users are taken from the public information on: `https://scratch.mit.edu/statistics/` (permalink: `https://perma.cc/4JEN-DYJD`)

[3] `https://scratch.mit.edu/discuss/` (permalink: `https://perma.cc/663D-RNEK`)

To build a dataset for our qualitative analysis, we generated a sample of 400 discussion threads about variables and lists from the Scratch discussion forums. Because we were interested in how kids learn to make projects with variables and lists by engaging in learning resources curated in discussions, we limited our sample to two subforums that emphasized question asking: *Help With Scripts* and *Questions about Scratch*. We chose to study forum threads instead of the more widely used project comments because previous work suggested that only a small number of comments were related to problem solving (Shorey et al., 2020a). The dataset that we used for sampling contains discussions that took place between October 11, 2012 and April 5, 2017.

To acquaint ourselves with our setting, we spent several weeks browsing the forums. As part of this process, we found that the many posters of threads about data structure did not use the specific terminology of "data block," "variable," or "list" in their posts. Therefore, to include a broad range of conversations about data in the Scratch forums, we followed advice from Trost (1986) to build a "statistically non-representative stratified sample" that would ensure that a range of different ways of talking about data were reflected in our data, but without concern for each type's prevalence. To do so, we sampled threads in three stages using different keywords. First, we randomly sampled 100 threads from titles containing the keywords "variable," "list" or "data." Second, we sampled 10 random threads each from the 11 most common variable or list names—as identified in an analysis of the Scratch code corpus by Dasgupta (2016)—for 110 threads total. These terms included "high scores," "lives," "inventory," "leaderboard," "speed," "timer," "counter," "words," "points," "velocity," and "answers." Third, to further increase diversity in our dataset, we randomly sampled two threads with each of the other top 100 variable and list names. This step resulted in an additional 200 threads. We only included threads with more than a single post because we wanted to ensure that the thread contained at least some form of a discussion. All together, these sampling steps resulted in 410 threads. Because 10 threads were included in more than one of our samples, we ended up with a total of 400 threads that contained 2,790 posts and 963,593 words of content—equivalent to 547 pages of single-spaced text.

We analyzed these data using Charmaz (2006)'s approach to constructing grounded theory. In the open coding phase, the first author led the analysis and started by annotating the threads with both line-by-line and incident-by-incident codes. In the process, the first author regularly shared coded data with the rest of the

team and iteratively update the codes based on team discussion. For example, in vivo codes were generated about what kids were creating using data structures: "making a score counter," "making a leaderboard," and so on. Following Charmaz, a small number of "sensitizing codes" drawn from existing theories were also used in the open coding phase, such as "engaging in collaborative debugging," (Shorey et al., 2020a) "helping through remixing," (Dasgupta et al., 2016b) and so on. In the axial coding phase, the first author led the process of code development by grouping initial codes into themes and meta-themes. For example, the initial codes about what kids were creating using data structures were grouped into an axial code called "functional use of data," with sub-level axial codes on variables and lists. This process involved repeated team discussions on code development, several rounds of iteration on code schemes, and re-coding data. Lastly, we composed memos to describe connections among the axial codes. Our findings reflect the content of the memos written about the major themes that emerged at the end of our analysis.

3.5.2 Findings

Our analysis resulted in three major themes. First, we discovered evidence that learners, driven by an interest in making specific popular game elements, tend to adopt a narrow set of functional uses of variables and lists. Second—and as a result of the first finding—we found that user-generated learning resources about variables and lists are framed around those specific examples of functional uses. Finally, we identified that those specific examples become archetypes that restrict the breadth of future functional uses in the community. Together, these findings describe a grounded theory about how interest-driven content creation can limit learning opportunities.

Learners create projects with functional uses of variables and lists specific to their interests in game-making

We found that Scratch users were often introduced to variables and lists when they engaged in discussions about specific functional elements of their projects. Because kids in our sample usually had specific goals for their projects in mind, but little knowledge about how to realize those goals in code, they would describe the particular thing they wanted to do when seeking help. The concepts of variables or lists would typically be brought up by someone else in response. Although a quarter of our sample were not selected on variable

names and half of our sample included threads based on 100 different variable names, game-making was an almost ubiquitous topic of discussion. Over and over, we observed kids phrasing their questions in terms of game-related goals in discussions in which variables and lists were eventually brought up. In the following sections, we discuss the canonical game-related use cases for variables, lists, and cloud variables in turn.

Variables: Two examples of common game elements that kids like to make in Scratch are *score counters* and *animations*. Score counters are a game element that keeps track of a player's score or remaining lives. Although this game element is broadly familiar to Scratch users, a user seeking to implement a score counter for the first time may not know about variables. Indeed, it is not always even obvious to users who know about the existence of data blocks in Scratch that a variable is an appropriate way to keep track of a changing quantity. Furthermore, it can be challenging for kids to implement a counter and integrate it into their program. For example, K1 asked: "I would like to know how to add lives in my game. I want it so that whenever the main character touches a ghost, it would lose one life." In interactions like these, other users would introduce variables and their functional use as a score counter. In this case, a reply from K2 suggested K1 "create a variable with the name lives" and use it to control the visual elements that represent the character's lives. Although carefully scoped to the specific problem faced by K1, the reply highlighted the role of variables in tracking changes.

Another common pathway to learning about variables involved animated objects in games. Animation frequently relies on variables because it involves changing the speed or size of objects when triggered by conditions. For instance, K3 asked:

> I'm making a pong game where I want to add a control to tell the ball to go faster. Is there a button for this? If not, how can I make this work?

K3 arrived to the Scratch forums knowing that they wanted to vary the speed of a pong ball. For them, the challenge was the specific case of making a ball move at a range of different speeds. Responding to their question, K4 pointed to variables:

> "Somehow, you must tell the ball to move. Make sure you use a variable. Like use 'move [speed] steps' rather than 'move 5 steps'. Then you just need to set your variable to the speed you like."

This response suggested that the K3 use variables and told them how. In these examples and many others, we saw that variables—both the concept and the term—almost never appeared in learners' initial questions.

Instead, many of the Scratch forum's learning resources about variables existed in answers to questions about score counters, animation, and other game elements.

Lists: We found a similar pattern for the list data structure. Frequently, kids were introduced to list data structures when trying to figure out how to make inventories—a game element with which players can store and retrieve items. For example, K5 asked: "Does anyone have an idea how to make a good inventory for a game?" K6 answered: "You can use the list block to store your items in the inventory." In another example, K7 said, "Allllllright so I'm making an inventory for game (who wouldn't want one?) so I don't know how to make one. Can anybody help?" These kids were all pointed to lists. Discussions like these helped kids who were struggling to build inventory features connect the abstract concept of a list with its functional use of storing multiple game items and played out repeatedly in the forums.

As with variables, kids imagined how an inventory would be used in the context of their games. For example, they described backpacks of weapons or a pool of correct answers in a quiz game. These kids also imagined rules describing how a player should manipulate items in the inventory. For example, K8 wanted to make a weapon inventory to hold "basic armour" and hoped to "make the player lose less blood when he/she has those items." After sharing these ideas, they received suggestions to put a list data structure within an "if else" statement. Cases like this suggest that building inventories allowed kids to learn not only about how to populate and read from lists, but also about basic list operations like deleting and appending items and about conditions and loops. In some cases, more advanced learners would describe methods for accomplishing complicated tasks imagined by novices:

> K9: "I am making a game where you can buy food and eat it. I want it so you can delete a certain food item from a list... so when you click, the sprite called 'Strawberry Popsicle' disappears, but also the name 'Strawberry Popsicle' disappears from the list too."
>
> K10: You need to search in the list and find the item that you want to delete. You can just look at each item in the list and compare it to the one you are looking for. Then you stop when you find it or get to the end of the list. This is called a sequential search. [Example code to solve the problem]

This thread shows how relatively sophisticated algorithms were explained in terms of very specific use cases, often with example code. By exchanging ideas about inventories in games, kids introduced each other to lists, their function, and the way they could be used.

Cloud Variables: A final example extends this pattern to *cloud variables* (see §3.3). Cloud variables'

ability to store data in ways that are persistent and shared were essential for users building "leaderboards" or "high score" systems that could record, rank, and display scores from multiple players—e.g., "a leaderboard in which the highest scores of every player of the game could be saved" (K11). Discussions about leaderboards often involve pointing out the existence of cloud variables, the differences between local and cloud data, and ideas about how to write code to use both. These conversations often segued into advanced programming topics like the encoding and decoding of strings. As with variables and lists, these conversations typically remained focused on the specific use case of leaderboards.

In all three cases, specific use cases became linked to specific data structures—variables with counters and animations, lists with inventories, and cloud data with leaderboards. Because questions tended to focus on these types of elements, discussions about solutions did as well. Through this process of user-to-user support, kids learned how to apply data structures in an informal and unstructured manner. As we describe in the next section, both these conversations and the games that Scratch users created acted as learning resources about variables and lists that were subsequently used by other learners in the Scratch community.

User-generated learning resources about variables and lists are framed around specific examples of functional uses

One feature of informal online learning environments is that conversations and solutions act as learning resources for subsequent participants facing similar challenges. In ways that are visible in our examples in §3.5.2, both the questions posed and the answers provided in our sample tended to focus on specific game-related functional uses. As a result, learning resources about how to use variables and lists tended to be framed in terms of specific game elements and rarely engaged with the more general concepts about data structures. For example, the following threads show a discussion on how to change the speed of a ball using variables:

> K13: "Can someone help me figure out how to change the speed of the ball when it hits the paddle a certain number of times?"
>
> K14: "You can make a variable called HITS, set it to 0, change it by 1 every time you hit the paddle. When it gets to the number you like, change costume and set counter to 0."

In this exchange, K14's answer details exactly what K13 should do to solve their problem—specified down to the names of variables. While this answer likely solved K13's problem, it is anchored on a very specific

functional use of variables and did not explain, or suggest that there existed, other potential functional uses. Discussions like this produce learning resources that are extremely specific to the question askers' use case.

When helping others use variables, kids would often refer to popular or example projects that contained working code. For instance, K15 asked "how to make a counter for scores using a sprite making clones of itself" and was directed by others to an established code chunk "changeScore method" in an existing project. In some cases, kids with more advanced knowledge would post snippets of working code:

> K16: "So I have a chat game, when "hello" is clicked, the robot would say "hello". I wonder how to make the robot say like a option of things such as "hey" or "yo" instead of "hello" all the time?"
>
> K17: "Put all the hello, hey words in a list then use this code: say (item (random v) of [list v])"

Although these are wonderful examples of kids mentoring each other, the solutions often offered by kids in our data were so specific—and almost always related to game elements—that they would be unlikely to help a novice learner build a conceptual understanding of data structures. It is not hard to imagine that, if a kid with a specific problem solvable with variables were to browse the discussion threads we analyzed, they might not be able to understand how variables could solve their problem unless they were making a game that was similar to one made by a person who had posted a question.

Furthermore, kids offered code might be able to use it without understanding it (Salac and Franklin, 2020). We saw many examples of kids requesting working solutions and many others who seemed happy receiving code that could be copy-and-pasted into their programs. For instance, K18 requested help in the form of an insertable code block: "Does anyone know how to make a smooth jump script? If you can make it into a custom block that would be great." In another example, K19 described the specific effect they wanted and expressed hope that someone could write the code for them:

> "I want to have a list that has these items: ham, cheese, egg, butter... I need it to find egg and read out it's number in the list. Is there a working script for this?"

As requested, K19's post was followed by a code snippet with variables named as K19 imagined them.

These examples are part of a broader pattern. To support kids like K18 and K19, the Scratch community creates solutions that are directly applicable to particular use cases. While these responses help kids overcome their problems quickly, directly workable solutions mean that kids might not see the broader picture of how variables and lists can be used. We explain in the next section how, because learning resources

in communities like Scratch consist of questions and solutions accumulated over time, this high degree of specificity in knowledge resources can result in difficult learning experiences for some.

Specific examples become archetypes and limit the breadth of functional uses in the community

Because learning resources were framed around specific examples of functional uses, many of the Scratch users in our forum dataset appeared to have a limited understanding of what variables and lists could do. This restricted understanding meant that even users without an expressed interest in making games, or particular elements in games, were presented with resources based on them. For example, a user expressing curiosity about data blocks in very open-ended terms received an answer that based on score counters:

> K20: "I think data blocks can be useful in my projects, but I don't know how to use them."
>
> K21: "You are saying variables and lists? For variables, they are just ways to name and store things. So if you make a game and want to keep the score then you'd create a variable called score."

Although K20 did not express any interest in games, K21's response was focused on them.

This reliance on canonical use cases was particularly obvious in discussions about more advanced cloud variables. For example, in the beginning of the following thread, K22 stated that they did not have knowledge about cloud variables. Despite lack of knowledge on the concept, K22 directly pointed to a canonical use case of it that they had heard of, that is, multiplayer games with leaderboards and high score lists. Immediately after this post, two other kids (K23 and K24) started a discussion thread about the particular use case:

> K22: "I have no idea what cloud variable is but I heard you could make multiplayer games with it."
>
> K23: "They are shared by 2+ instances of your game. If you make a very simple game in which you add 1 to a cloud variable when a sprite is clicked. Save your game. Then open the game in two new windows in your browser."
>
> K24: "They are usually for High Scores and Multiplayer Games."

Although K22 signalled that they were open to exploration, the answers they received from K23 and K24 indicated the most canonical goals and interests around cloud variables.

We found that kids with interests that deviated from canonical use cases had difficulties finding learning resources that fit their interests. For instance, when K25 posed a very general question about "how to use cloud variables to save data from users" in a more general thread about cloud data, other kids tried to help

by posting examples of a high score system that involves cloud variables. K25 expressed confusion because the solution for a high score system did not fit their own goals and said, "but my game isn't one of those scoring games. I want to make a storyline game." K25's comment revealed what much of the Scratch forums community takes for granted. K25 ended up not receiving help in the thread. Over time, their post was lost and ignored in a stream of more typical messages about leaderboards.

In summary, we found that certain functional uses became the Scratch forum's go-to examples for explaining variables and lists. Because learning resources were built cumulatively, it is not hard to imagine that more projects with these functional uses of variables and lists would be created over time. These new projects, in turn, became new learning resources as well. Learners who wanted to explore different use cases had fewer relevant resources to guide them.

3.5.3 Synthesis: A theory of social feedback loops in interest-driven online learning communities

Our findings echo Papert (1993)'s emphasis on "personally powerful ideas" and Ito et al. (2013)'s description of interest-driven, community-based learning. We found that kids in Scratch are motivated to use variables and lists to explore their passions and that they leverage content shared by others in the community to do so. Because learners are working with specific goals in mind, they run into the need for variables and lists while trying to implement specific elements. They are tutored by peer-produced learning resources framed in terms of those specific functionalities. In this sense, our finding contributes to both the literature of computational participation and computing education by describing that interest-driven content creation can be a potential pathway to support learning of functional uses of complex computing concepts.

At the same time, we also discovered a unintentional side effect of this type of learning. We identified that because community-generated learning resources in the form of Q&A, tutorials, and project examples tend to be directed toward specific functional uses, those that represent common interests can become archetypes in ways that leave less room for unconventional interests. In some cases, it can also lead to a shallow understanding of underlying concepts (Salac and Franklin, 2020). In our sample, the almost exclusive focus on certain game elements raises concerns about whether learners who are not interested in making these elements will be well served by community-generated resources. Furthermore, we observed that kids

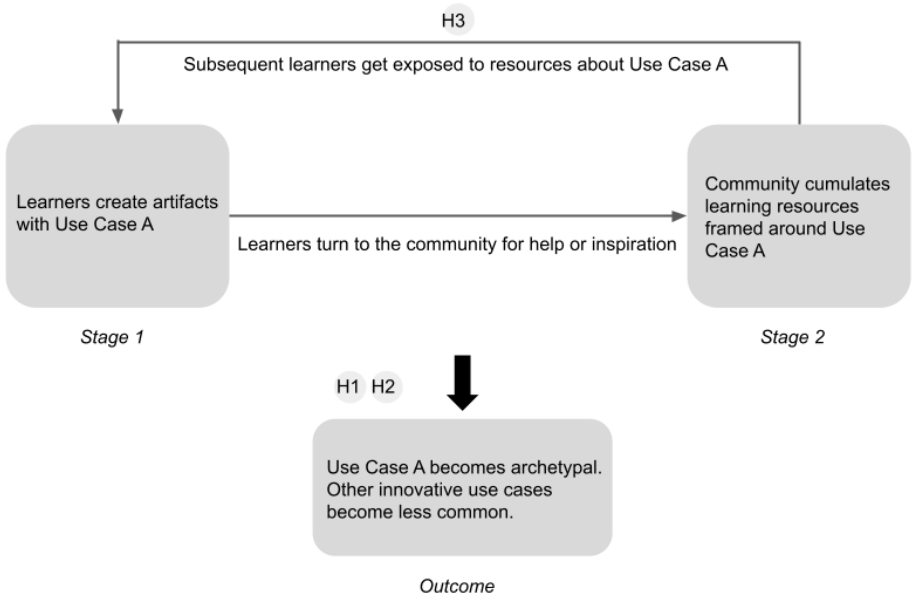


Figure 3.2: Hypothesized social feedback loop in interest-driven online learning communities

who were open to exploring different functional uses are pointed to the archetypal use cases. Overtime, this practice can make the most common use cases even more archetypal.

Inspired by the existing body of literature on network-based and social processes that lead to increased concentration of resources over time, such as the Pareto principle (Juran, 1954), preferential attachment (Barabási and Albert, 1999), and the Matthew effect (Merton, 1968), we theorize that this process plays out as a *social feedback loop*. Our social feedback loop theory is situated in the specific context of interest-driven computational learning in Scratch and can be summarized as follows: *interest-driven content creation can result in certain types of creation becoming archetypes that make community-generated learning resources more homogeneous and support an increasingly limited set of learner interests over time.*

This theory is visualized in Figure 3.2. Stage 1 suggests that some types of creation (Use Case A) will be more popular than others and that more users will tend to create artifacts with Use Case A than other use cases. This may be due to initial user base homogeneity, targeted recruiting in the beginning of the community, examples used in documentation, and so on. The arrow pointing from Stage 1 to Stage 2 captures the process of learners running into problems and seeking community support. We argue that users will tend to ask questions framed specifically around Use Case A and draw inspiration from others' artifacts with Use Case A. Stage 2 shows the results of this process. As learners successfully receive support, they

produce new artifacts with Use Case A that can serve as learning resources for others. The arrow on the top of Figure 3.2 pointing from Stage 2 back to Stage 1 shows how subsequent learners draw inspiration and support from these accumulated learning resources and, as a result, become even more likely to create artifacts with Use Case A in the future. The box on the bottom of Figure 3.2 captures the outcome of the feedback loop based on our findings from §3.5.2. As Use Case A becomes a archetype, the community's collective learning resources are increasingly focused on Use Case A as well. Learners like K25 in §3.5.2 who have different interests receive less support. Over time, innovative use cases will become less prevalent.

3.6 Study 2: Theory testing

Study 1's ultimate findings are a set of untested propositions. In a quantitative follow-up study, we conduct tests of three hypotheses that we derived from the theoretical model presented in §3.5.3 to begin the process of validating the theory. Our first two hypotheses (marked as "H1" and "H2" on the bottom of Figure 3.2) focus on the outcome of the feedback loop, that is, what we will observe if we assume the social feedback loop is occurring. In both cases, the hypotheses are that such a feedback loop will make use cases increasingly homogeneous over time. First, we hypothesize that certain genres of projects involving simple data structures will become more popular over time relative to other genres. Based on our findings in Study 1, we hypothesize that **(H1)** *over time, more projects involving variables and lists will be games*.

Second, we hypothesize that popular functional uses of variables and lists will be even more common relative to others over time. In Scratch, users are able to enter a free form string as the name of their variables and lists. Based on our observations in Study 1, users tend to name variables and lists as the specific things they are trying to make. Therefore, we treated the names that users assign to the variables and lists as a proxy to the functional uses. We thus hypothesize that **(H2)** *the names that users give to variables and lists will become more concentrated over time*.

While these hypotheses reflect what we would expect to see in aggregate if the hypothesized social feedback loop were occurring, our third hypothesis (marked as "H3" on Figure 3.2) attempts to capture part of the theorized mechanism, that is, users who get exposed to archetype use cases will create similar artifacts. Therefore, we hypothesize that **(H3)** *users who have been exposed to projects involving popular variable and list names will be more likely to use those names in their own projects compared to users who*

have never been exposed to such projects.

3.6.1 Data

To conduct our quantitative analyses, we used the *projects*, *project_strings*, *project_text* tables from the publicly available Scratch Research Dataset (Hill and Monroy-Hernández, 2017). For testing H3, we utilized one non-public column that records which users had downloaded others' projects. We restrict our analysis to projects created between September 2, 2008 and April 10, 2012 because affordances around data blocks were consistent during this period.[4] The period is earlier than the time window used in Study 1 based on differences in the datasets we had access to. For analytical simplicity, we decided to only include projects with variables and lists written in English. Finally, we restricted our analysis to *de novo* (i.e., non-remix) projects. This resulted in 241,634 projects that contained one or more variables authored by 75,911 Scratch users, and 26,440 projects that contained one or more lists authored by 12,597 users. We created both project-level and user-level datasets with a range of metadata available in the Scratch Research Dataset (Hill and Monroy-Hernández, 2017). In the spirit of open science, we have placed our full source code for dataset creation and analysis into a public archival repository.[5]

3.6.2 Analysis and Measures

To test **H1**, we used our project-level dataset to assess whether there is an increase over time in the proportion of games with at least one variable/list. To ensure that our assumption of games being a predominant genre of project was correct, we randomly subsampled 100 projects with variables and 100 projects with lists. Two coders classified these projects as "game" or "non-game" and reached high inter-rater reliability (Cohen's $\kappa = 0.88$). We found that 65% (CI = [54%, 74%]) of projects with variables and 52% (CI = [41%, 62%]) with lists were games.[6] This reinforces our sense, developed in Study 1, that games are the dominant genre of Scratch projects containing variables and lists. It also gives us confidence in our decision to use a measure of the prevalence of games over time to test our theory related to popular genres of projects.

Because it is difficult to manually identify games in our large dataset of projects, we define projects as

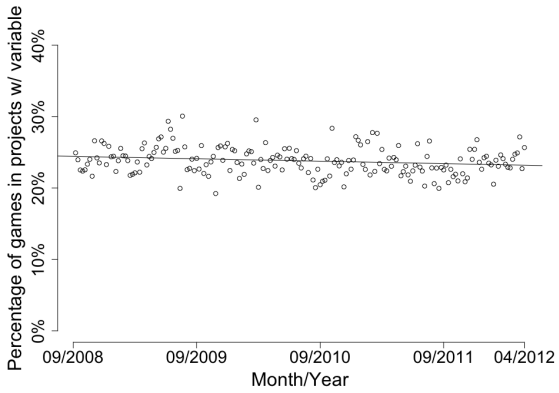
[4] `https://en.scratch-wiki.info/wiki/Scratch_1.3` (permalink: `https://perma.cc/FL57-FVNS`)
[5] `https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/2TRZ9N`
[6] Numbers within brackets are 95% confidence intervals computed using Yates' continuity correction.

games if they contain the string "game" or "gaming" in their titles or descriptions. To validate this measure, we again hand-coded samples of projects from four random samples: 100 projects with variables and at least one of the strings, and 100 similar projects without the strings; two similar samples of 100 projects with lists. The same two coders coded all 400 projects as game or non-games. Among the projects that contain variables, we found that 88% (CI = [80%, 93%]) projects with strings "game" or "gaming" were games, while only 48% (CI = [38%, 58%]) projects without those strings were. We found a similar pattern among projects with lists, where and 85% (CI = [76%, 91%]) and only 31% (CI = [22%, 41%]) projects were games, respectively. In other words, our method of identifying games using the strings "game" and "gaming" is high precision and somewhat low recall. Because our goal with H1 is to study change over time rather than baseline prevalence, low recall is not problematic as long as it is consistent over time. Analysis in Appendix A suggests that these proportions were consistent over the period of our study.

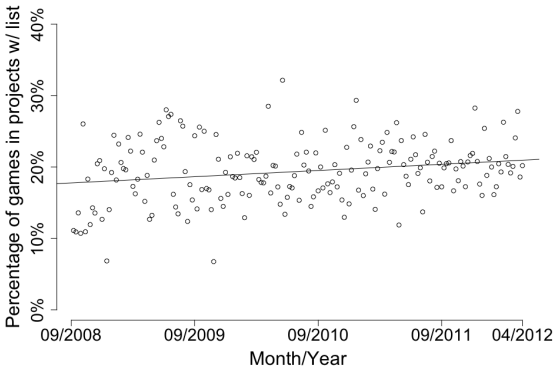
To test **H1**, we perform a logistic regression on the odds of a project with variables/lists being described as a game where the date in years in which the project was created is our independent variable. We include month-of-year fixed effects to control for seasonality. We used a linear specification of time because exploratory data analysis indicated that curvilinearity was unlikely a major concern.

To test **H2** that there will be increasing concentration in variable/list names over time, we operationalize "concentration" as the Gini coefficient of the distributions of variables across names for each week of data we collected (Ceriani and Verme, 2012). Originally invented to measure wealth inequality in a nation, Gini coefficients range from 0 representing perfect equality (if every variable name is used in an equal number of projects) to 1 reflecting perfect inequality (if only one variable name were used). We perform a linear regression using Gini Coefficient as our dependent variable and the same month-of-year fixed effects we use in H1 to control for seasonality. We use a linear specification of time for the same reason we do in H1.

H3 seeks to test the effect of exposure to popular variable/list names on subsequent behavior. We treat popular names as the 20 most frequently used names for variables or lists. Because it is not possible to measure exposure directly, we use a measure of whether a user has downloaded a project with popular variable/list names as a proxy. We feel this is justified because the only way to access the source code of a Scratch project during our data collection period was to download it. We use these measures in Cox proportional hazard models (Singer and Willett, 2003). Originally developed in epidemiology, we follow



(a) Percentage of games among projects with variables.



(b) Percentage of games among projects with lists.

Figure 3.3: Percentage of games among projects with variables or lists, per week, from September 2008 to April 2012. Lines reflect bivariate OLS regression lines.

the framework used by Dasgupta et al. (2016b) who used Cox models to measure online informal learning of computational thinking concepts in Scratch. Our models estimate the chance that a user in our dataset will share a *de novo* project with a popular variable name for the first time as a function of the number of *de novo* projects they have previously shared.

Our question predictor is a time-varying dichotomous measure of whether the user has downloaded a project with a popular variable name during our period of data collection. We conducted the same analysis for lists. Finally, we include a control variable for the total number of downloaded projects to capture overall exposure to other projects in Scratch—a potential confounder.

3.6.3 Results

The results from our hypothesis tests provide broad but uneven evidence in support of our theoretical model in §3.5.3. Figure 3.3a shows that, contrary to H1, the percentage of games in projects with variables decreased slightly over time. The hypothesis test shown in Table 3.1 suggests that this weak relationship is statistically significant ($\beta = -0.02$; SE < 0.01; $p < 0.01$) and that each year is associated with odds that are 98% the odds of the year before. On the other hand, our results for lists are in the hypothesized direction. Figure 3.3b shows that the percentage of games among projects with lists has been increasing over time. The results of our logistic regression in Table 3.1 suggest that this relationship is statistically significant ($\beta = 0.06$; SE $= 0.02$; $p < 0.01$). The model estimates that the odds that a newly created

	Variable	List
(Intercept)	-1.09^*	-1.62^*
	(0.02)	(0.06)
Year	-0.02^*	0.06^*
	(0.00)	(0.02)
Month fixed effect	yes	yes
Log Likelihood	-132558.53	-13058.12
Deviance	265117.06	26116.23
Num. obs.	241634	26440

$^*p < 0.001$

Table 3.1: Logistic regression models for the likelihood of a project including the term "game" or "gaming" in its title or description. Models are fit on two datasets including all non-remix projects containing variables ($n = 241,634$) and all non-remix projects containing lists ($n = 26,440$) from September 2008 to April 2012.

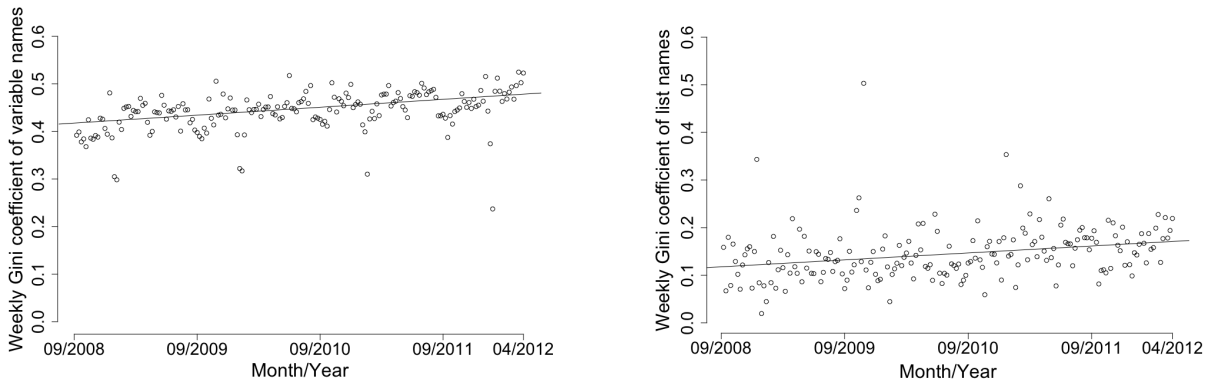
	Variable	List
(Intercept)	0.38^*	0.11^*
	(0.01)	(0.01)
Year	0.02^*	0.01^*
	(0.00)	(0.00)
Month fixed effect	yes	yes
R^2	0.38	0.13
Adj. R^2	0.34	0.07
Num. obs.	190	190

$^*p < 0.001$

Table 3.2: OLS time series regression models on the Gini coefficient of variables across variables names for all projects shared in Scratch each week ($n = 190$).

project involving a list is a game are increasing by 106% year-over-year. For instance, the model-predicted probability of a project with lists created in March 2012 being a game is 22.3%, while that of a similar project created in March 2009 is 20.8%. This estimate translates into 491 more games than we would have expected if there had been no year-over-year increase. We also checked the percentage of games among all projects (not limited to those with variables or lists) over time and found there was no obvious change in the overall game percentage. The details of this analysis is in the Appendix A. In other words, our findings for H1 align with our expectation on the outcome of the hypothetical social feedback loop for lists, but not for variables.

We found strong support for H2 that variable and list names would become more concentrated over time.



(a) Weekly Gini coefficients of variable names.

(b) Weekly Gini coefficients of list names.

Figure 3.4: Weekly Gini coefficients of variable and list names over time. Lines reflect bivariate OLS regression lines.

Figure 3.4a shows differences in Gini coefficients over time for variables and Figure 3.4b shows the same measure for lists. Both figures clearly show increasing concentration. Hypothesis tests from OLS time series regression models are reported in Table 3.2 and reveal that these relationships are statistically significant for both variables ($\beta = 0.02$; SE < 0.01; $p < 0.01$) and lists ($\beta = 0.01$; SE < 0.01; $p < 0.01$). We estimate that the concentration across variables has increased from a Gini coefficient of about 0.41 in 2008 to 0.50 in 2012. For reference, this difference is similar to the difference in concentration of wealth between the United States (Gini coefficient = 0.41), which is more concentrated than 68% of countries globally, and Zimbabwe (Gini coefficient = 0.50) which is more concentrated than 90%.[7] In other words, the distribution of variables names is both quite concentrated and is increasing in concentration over time. Although list names are much less concentrated in general, they are increasing in concentration at a similar rate. Our findings for H2 provide additional support for what we expect to see in the community if the hypothetical social feedback loop is occurring.

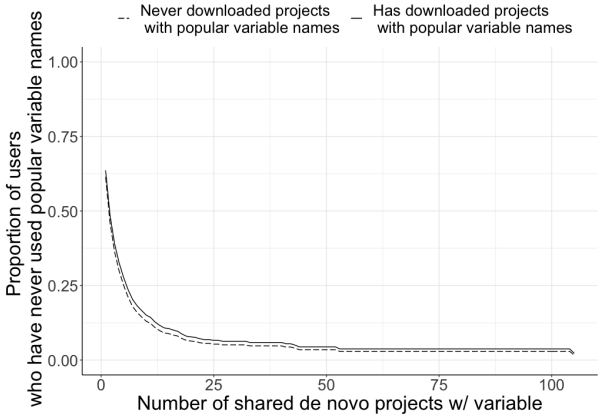
Table 3.3 shows parameter estimates from our Cox models and shows mixed support for H3. Although we hypothesized a positive relationship between exposure to and subsequent use of popular variable names, we find that our measure of exposure to popular variables is associated with a risk of using them that is only 93% as high ($\beta = -0.07$; SE $= 0.01$; $p <= 0.001$). On the other hand, users who downloaded projects with popular list names are more likely to use those names in their own projects than those who did not.

[7] `https://data.worldbank.org/indicator/SI.POV.GINI` (permalink: `https://perma.cc/CF58-34KQ`)

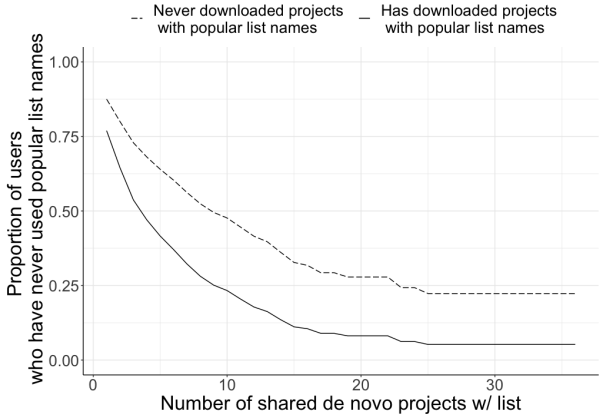
User	Risk of Using Top Variable Names	Risk of Using Top List Names
Downloaded projects w/ top variable/list names	-0.07^*	0.68^*
	(0.01)	(0.04)
log(# of 100 downloads)	-0.12^*	-0.07^*
	(0.02)	(0.02)
R^2	0.00	0.02
Max. R^2	1.00	0.97
Num. events	52967	3790
Num. obs.	88327	17869

$^*p < 0.001$; $^{**}p < 0.01$; $^*p < 0.05$

Table 3.3: Fitted Cox proportional hazard models that estimate the "risk" that a Scratch user will share a de novo project that uses a popular variable or list name for the first time, based on whether the user has downloaded a project with top variable or list names. Number of downloads is a control to capture general exposure to other projects in Scratch. A positive coefficient means increased "risk", while a negative coefficient means decreased "risk".



(a) Model-predicted probability of having shared a project with a popular variable name for two prototypical users who have/have not downloaded such a project.



(b) Model-predicted probability of having shared a project with a popular list name for two prototypical users who have/have not downloaded such a project.

Figure 3.5: Plots of model predicted estimates of the proportion for several prototypical users. In Figure (a), estimates are shown for two prototypical users: (dashed) a user who has never downloaded projects with popular variable names, and (solid) a user who has downloaded projects with popular variable names. Figure (b) is the same plot but for lists instead of variables.

The instantaneous risk of sharing a project with a popular list name for a user who downloaded at least one project with a popular list name is 1.97 times higher than another similar user who has never downloaded such a project ($\beta = 0.68$; SE $= 0.04$; $p < 0.001$). The small negative effect associated with variables may

be due to the fact that variables are used much more often than lists in Scratch so that kids may simply have more opportunities to be exposed to popular variable names.

Because Cox models are difficult to interpret, we present visualizations of model-predicted estimates in Figure 3.5. Each panel includes two lines reflecting prototypical community members who downloaded one project before sharing projects with others: one prototypical user who had downloaded a project with a popular list or variable name; the other who did not. Figure 3.5b shows that members who have previously downloaded a project with at least one popular list name are more likely to use a popular list name in their own subsequent projects. Figure 3.5b shows that our model predicts that ~50% of users who have shared 10 *de novo* projects and who have never downloaded projects with popular list names will have never used a popular name in their own projects, while only ~25% of similar users who have downloaded such projects will not have. Although the negative effect is statistically significant, we do not see a similar effect with variables in Figure 3.5a. In summary, our findings for H3 offer some evidence for the mechanism of the social feedback loop—one again for lists but not for variables.

3.7 Discussion: Challenges & Opportunities of supporting learning through interest-driven content creation

While researchers have long argued that interest-driven participation can allow learners to explore and be creative (Kim et al., 2017a; Chan et al., 2016; Marlow and Dabbish, 2014b; Monroy-Hernández, 2007), our case study on computational learning in Scratch indicates that this type of participation might also create self-reinforcing social processes that constrain community imagination around advanced subjects. In our first study, we use data from the Scratch forums to build a grounded theory describing a feedback loop that exists between learners' interests and the resources they create. This loop makes it easier for some users to learn about variables and lists—but in ways that are increasingly focused on a set of specific functional uses that have been used by others extensively in the past. We test several hypotheses derived from this theory in a series of quantitative analyses of the Scratch code corpus and find broad, if uneven, support for the theory.

Our study contributes to the literature on computational participation by highlighting a trade-off between interest-driven participation and learning about computational concepts. On the one hand, our study shows

that novice learners can learn functional uses (diSessa, 1986) of advanced computational concepts by engaging in discussion and social support. On the other hand, we found that such learning can be superficial and not conceptual or generalizable. Echoing prior studies that raise concern about the lack of depth in the most common forms of computational participation (Shorey et al., 2020a; Gan et al., 2018), our study argued that learners' preference for peer-generated learning resources around specific interests can restrict the exploration of broader and more innovative functional uses. Although it is conceivable that a narrow set of archetypal use cases could be beneficial for learning for some, increasingly homogeneous use cases stand in clear opposition to the common design goal of broadening participation.

While our empirical evidence is limited to Scratch, we speculate that our theory describes a common dynamic in informal learning environments. In the rest of this section, we discuss three challenges for Scratch and broader online learning communities implied by our theory: (1) a decrease in the diversity of resources that novice learners might draw inspiration from; (2) privileging participation by learners with the most common interests; and (3) a lack of understanding of concepts that cover broad functional use. We argue that each challenge represents promising opportunities for design.

3.7.1 Limited sources of inspiration

It has been argued that informal learning systems should offer "wide walls"—affordances that support a range of possibilities and pathways with which learners can creatively construct their own meanings (Resnick and Silverman, 2005; Papert, 1993). In the context of Scratch, previous research suggests that novice learners show increased engagement when the walls are "widened" through new design features (Dasgupta and Hill, 2018a). Our findings describe how the unstructured nature of the Scratch online community can lead to overrepresentation of certain ways of applying knowledge—effectively "narrowing" the walls.

A range of common social features presented in Scratch and similar online artifact curation and Q&A communities are likely to reinforce this dynamic. For example, up-votes and likes may externalize the popularity of certain posts (Agichtein et al., 2008a), artifact sharing can draw attention to already-visible topics (Cheng and Zachry, 2020a), and gamified rewards can incentivize already-popular styles (Cavusoglu et al., 2015a). In each case, these features may make it difficult for learners to see beyond the limited

set of popular use cases that the rest of the community is presenting. This narrowing is clearly unintended. Learners interested in a common application of a concept produce long discussion threads and an abundance of examples and tutorials out of a real desire to help. And indeed, these examples frequently *will* help others.

Inspired by the call made by Buechley and Hill (2010), we suggest that future online informal learning communities should offer affordances that empower learners to leverage the "long tail" of novel use cases. Designers of the platforms should seek to help learners recognize new use cases and examples. For instance, designers might highlight novel or unusual projects and provide recognition and status to community members engaged in unconventional creation. For example, the Scratch front page has a curated section designed to showcase projects which could serve this purpose. Adaptive recommendation systems might help learners broaden their sources of inspiration by directing them to topics and genres that are different from what they are familiar with. Community moderators might guide conversations toward novel perspectives when there has been enough discussion of similar ideas.

3.7.2 Narrowed opportunities for participation

A related challenge is that increasingly homogeneous use cases might marginalize learners not interested in popular topics in ways that lead to demographic inequality. A number of studies in computational learning communities have shown that the underrepresentation of learners' interests and identity may give rise to a sense of being excluded or marginalized (Buechley and Hill, 2010; Ford et al., 2016; Richard and Kafai, 2016). For instance, although many girls use Scratch, there is evidence that girls are generally less interested in making games than boys using the platform (Funke and Geldreich, 2017; Hsu, 2014). As a result, game-specific learning resources related to data structures may make it easier for boys to learn about data. In that HCI researchers have built curriculum around game-making in Scratch as a way of building up computational thinking (Troiano et al., 2020), we are concerned about the implications of this approach for users—disproportionately girls—who are not interested in making games.

As a possible step to address this challenge, community designers might elicit users' interests and connect users with similar interests to each other (Kraut et al., 2011). The community might also match learners with different backgrounds and interests and motivate them to exchange examples and feedback. Moderators could also offer more support and resources to users who want to explore less popular genres. For example,

they might connect users with unusual interests to experts in the community. In the past, the Scratch online community has had a "welcoming committee" designed to help newcomers get started (Roque et al., 2013). Our findings suggest a potential way to target these sorts of efforts.

3.7.3 Confined understanding of broader knowledge

The final challenge involves helping learners acquire an understanding of underlying computational concepts that goes beyond specific use cases. Our findings are consistent with the broader literature on learning and creativity suggesting that when a group of people engage in creative activities, they will generate less diverse ideas after having been shown popular examples (Yu and Nickerson, 2011a; Kulkarni et al., 2014). Our findings also echo prior work that suggests although the ability to remix can provide inspiration and scaffolds (Dasgupta et al., 2016b), there may be tradeoffs in terms of originality and whether learners might struggle to acquire transformative programming skills (Hill and Monroy-Hernández, 2013). Our study further suggests that although community-produced examples may grow in volume over time, they may only represent material for an increasingly narrow set of functional uses. Informal scaffolds like discussion messages and unregulated artifact catalogs may not always help learners see the big picture or master a skill.

We believe that this challenge points to a final opportunity for learning resource exploration and search systems that focus less on specific examples. For instance, hierarchical tagging and grouping mechanisms could be designed to help novice learners understand the relationship between specific examples and higher level concepts. In Scratch, a high-level collection could be called "use cases of data structures," and the subcategories could include games, story-driven projects, and artistic media. Additionally, the discussion forum could be seeded with prompts to support the identification of underlining conceptual knowledge and to explicitly connect examples with human mentoring (Ford et al., 2018a), cognitive apprenticeship (Hui et al., 2018a) and automatic annotation (Chan et al., 2016).

3.8 Limitations

First, the interest-driven content creation investigated in this chapter is limited to the setup of Scratch forums and projects. For example, Scratch users are mostly children, use block-based programming interface provided by Scratch, and tend to make programs with two-dimensional media. Other computational learning

platforms or informal learning communities targeting a different subject may not include these features, and we cannot know exactly how our theory and findings will translate to other contexts. Similarly, since our theory is specifically built around variables and lists, we cannot know how it can be generalized to other type of computational learning and informal learning in general. Although in the discussion section we proposed design implications for online learning communities in general, these implications are merely speculative. Our main contribution is a case study of the Scratch community and we can know when our theories will generalize to other informal learning contexts. We share our work with the hope that future scholars will build on and critique our work by testing these theories in the communities that they study.

Scratch is used in many languages (Dasgupta and Hill, 2017a). Our work is limited in that it only considers English language content. We do not know what impact the multi-lingual nature of Scratch has on our analysis or if the dynamics we observe are also present in other linguistic subcommunities in Scratch. Our strategy to detect project genre in our test of H1 is limited by language in that not all games have the words "game" or "gaming" in their title or description and some non-game projects do. Additionally, Scratch users learn from resources including project comments, tutorials, and one-to-one mentorship both within and beyond the Scratch community. In that forums are not the sole (or even primary) way that kids learn in Scratch, Study 1 might be missing important social dynamics in other places.

As we discussed in §3.5, our sample in Study 1 is nonrepresentative in ways that may shape our findings. Because a quarter of our sample selects on the 11 most popular variable/list names—and because these names mostly indicate game elements—our qualitative dataset may be skewed toward game-making in ways that shape our findings in Study 1. The random samples and population-level data used on Study 2 is an effort to address this issue.

Another set of limitations stems from our reliance on imperfect proxy measures in Study 2. For example, we use downloads as a measure of exposure to test H3 because downloading was the only way to view the source code of a Scratch project before 2013. That said, users might download projects to deal with a slow internet connection or for a range of other reasons.[8] Although we feel confident that downloads will be correlated with exposure, we have no way of knowing why a user downloaded any given project. Similarly, we used user-defined names of variables and lists as a proxy of the use cases of Scratch data structures.

[8] `https://en.scratch-wiki.info/wiki/Project_Downloading#Benefits_of_Downloading_Projects` (permalink: `https://perma.cc/FAM2-L3SP`)

Although our findings from Study 1 suggest that variable and list names largely represent what users were making with variables and lists, we cannot know for sure if this is accurate in every project and there is chance that some users may use names that are inconsistent with the use case. In addition, like most other studies of informal learning online, we can only observe learning experiences and not outcomes. Measuring learning outcomes in a community like Scratch is difficult because learners arrive with different interests and aspirations and take different paths. Although we measure the presence or absence of variables and lists in projects, we can not know whether they are being used correctly or whether project creators understand the code they write (Salac and Franklin, 2020).

Finally, although we theorize that there is a causal link between our proposed social feedback loop and increased homogenization of community-produced learning resources, we present no causal evidence. For example, our test of H3 provides evidence of a correlation between exposure to popular list names and an increased likelihood of future use of those names. This relationship might also be due to variables that are correlated with, but not caused by, a social feedback loop like the one we describe. Similarly, we try to argue that the narrowing trend in the usage of variable and list names that we discovered is an indicator of the social feedback loop that narrows the creativity in the community. However, there may be other factors outside the community, such as pop-cultural trends and school education, that contribute to this narrowing effect. To summarize, the evidence we present in Study 2 should be interpreted as similar to what we would expect to find if our theory were true—nothing more.

3.9 Summary

This study suggests that the way programming concepts related to data are understood in informal environments diverges from the traditional school-based learning. A key takeaway is that informal learning communities are able to support the bottom-up comprehension of programming concepts around data through examples and use cases that resonate with the members' passions. In this example of the Scratch Online Community, the community collectively establishes the previously challenging functional use cases of variables and lists. This knowledge-building process is grounded on community members' abilities to articulate specific needs of data for their projects and to create and unitize artifacts that reflect the common interests and language in the community, fostering broader community engagement with data.

However, this collection of peer-produced learning resources does not necessarily lead to a comprehensive understanding of the programming concepts. Indeed, such learning can sometimes be superficial and lacking a deep conceptual foundation. A learner's preference for peer-generated resources tailored to specific interests might limit the exploration of wider and more innovative applications. Such a focus might overlook niche interests, potentially leading to reduced creative diversity over time. This trend goes against the very essence of informal learning communities and needs to be recognized and addressed in the design to facilitate learning of programming and other technical skills around data. To mitigate this, informal learning platforms must introduce policies and features that counterbalance the effects of the social feedback loop we have identified. Potential strategies include promoting a diverse range of inspirations, encouraging participation from members with varied interests, and scaffolding the construction of knowledge that is broader and more universally applicable.

Chapter 4

Study B: Kaggle

Prologue

This chapter delves into Study B, conducted between 2019 and 2020, roughly around the same time as Study A. While exploring how informal settings influence the development of programming skills related to data, I came across studies focusing on professional data scientists, which indicate that data science efforts often involve collaboration, feedback exchange, and resource sharing among multiple stakeholders (Zhang et al., 2020). Interestingly, collaborative and communication skills, while potentially key to data literacy, tend to be overshadowed in studies that emphasize curricula and classroom settings. Thus, I became curious about the challenges and opportunities regarding collaboration around technical engagement with data in informal environments.

This prompted my investigation of the Kaggle online community in Study B. At the time of this study, Kaggle stood out as the largest platform for collaborative data science problem solving. Members of the Kaggle community collaborate on data science problems through shared computational notebooks and online discussions around the notebooks, where they communicate the process of data analysis through a combination of code, text, and media. The platform also provides a structure that explicitly presents participant rankings and varying levels of experience and achievement. This structure provided a valuable lens for me to study the strategies adopted by novices and more experienced members, observing the competencies, practices, and challenges involved in the collaboration within and across experience levels. Moreover, Kaggle

by its nature is a platform for open innovation contests that hosts data science competitions, adding another layer that enables me to gauge the impact of its competitive social dynamics on collaborative behaviors.

In this particular study, I conducted 14 interviews with Kaggle community members, investigating how data scientists of varying experience levels engage with the shared computational notebooks that document the data analysis process and the online discussions around them. I unpacked the differential impact of community characteristics and dynamics on both novice and experienced data scientists. At the end of this study, I discussed the necessity of recognizing and catering to the different communication and collaboration practices and needs of community members across different levels of expertise in working with data in informal settings.

The following sections of §4.1 to §4.8 in this chapter feature the paper I published at the ACM SIGCHI Conference on Computer-Supported Cooperative Work & Social Computing in 2020 (CSCW'20) (Cheng and Zachry, 2020a) The paper was a collaborative effort between myself and Dr. Mark Zachry, in which I was the lead of the work and was responsible for the design of this study, data collection and analysis, and manuscript writing. The content of the paper remains in its original form without modifications. In §4.9, I included a short summary of the lessons learned from this study with respect to the broader thesis of this dissertation.

4.1 Introduction

Nowadays, it is becoming increasingly common for people to build community knowledge with each other in open online systems (Scardamalia, 2002) such as social Q&A (Tausczik et al., 2014b), open source projects (von Hippel and von Krogh, 2003) and online creative communities (Xu and Bailey, 2012b). In these systems, participants collectively advance knowledge that is publicly accessible to the community through distributed contributions, including sharing the artifacts they have created, exchanging feedback, initiating discussions, etc. Such activities are crucial to both the quantity and quality of user generated content, promoting self-regulated learning among members (Greenhow and Robelia, 2009; Gray, 2004; Fiesler et al., 2017c) and growth of the community (Kraut and Resnick, 2012). Yet, as demonstrated in multiple CSCW and HCI studies, maintaining sustainable, high quality contribution among community members is challenging (Kraut and Resnick, 2012; Ardichvili, 2008; Nielsen, 2006) due to a variety of

reasons such as lack of motivation (Nielsen, 2006) and social barriers (Marlow and Dabbish, 2014a; Foong et al., 2017).

To address such challenges, a specialized type of knowledge building system — online open innovation contests — has recently emerged. Such contests are designed and conducted by organizations to broadcast difficult challenges to a large crowd and to leverage the collective intelligence of many people to generate new ideas and innovations. These contests have proven to be an effective model for collecting innovative knowledge artifacts (Chesbrough, 2003; Horton and Chilton, 2010) such as computer programs (Archak, 2010; Lakhani et al., 2010; Nag et al., 2012), designs (Hutter et al., 2011; Yang et al., 2009) and data science models (Tausczik and Wang, 2017b). Competitive mechanisms, including tangible rewards (e.g., monetary prizes) granted to the best solutions and gamified reputation systems (e.g., contributor rankings and achievement medals), are common features in these contests. Most interestingly, in addition to knowledge artifacts submitted by individual participants to the contests, researchers have observed lasting public knowledge building activities in such competitive communities, including public sharing of in-progress solutions (Tausczik and Wang, 2017b) and open discussions around ideas (Hutter et al., 2011). These public sharing behaviors seem counterintuitive as they may lower individual competitors' chances of winning, yet they are surprisingly common in these competitive systems.

Therefore, we see open innovation contests as a potential new design model for increasing contribution and community knowledge building in online systems. In particular, we contend that system developers who seek to borrow the design of contests for use in their knowledge building systems should have a deeper understanding of the behaviors that such contests encourage and support. While previous research effort has recognized and described public knowledge building behaviors in online contests, few have investigated why competitors contribute to public knowledge when such contributions may harm their own success. Even less is known about how competitive mechanisms affect knowledge building behaviors among different levels of participants. Our study therefore is guided by the following exploratory research question:

How and why do participants with different levels of domain experience contribute to and consume shared knowledge in online competitions?

To answer this question, we conducted in-depth interviews with 14 users of Kaggle Competitions,[1] the

[1] `https://www.kaggle.com/competitions` (permalink: `https://perma.cc/FSK7-LJS3`)

world's largest data science open contest platform, wherein participants compete for best solutions and also build knowledge through public code sharing (Notebooks) and engaging in public discussions (Tausczik and Wang, 2017b). In our extended, semi-structured interviews, we asked about users' motivations and experiences related to activities such as sharing and using directly executable notebooks, contributing questions and ideas to discussions, and finding teammates to collaborate with. We found that participants at the two extreme ends of data science experience, namely, the *experts* and the *beginners*, have been very differently impacted by the competitive mechanisms in their engagement in building and using public knowledge. Interestingly, while Kaggle Competitions has been viewed as a successful and open platform for data science enthusiasts to showcase ideas and learn from each other, we found that such opportunities are not equally accessible to experts and beginners. While competitive mechanisms motivate experts to share knowledge, they also lead them to form niches and share solutions that are not readily understood by beginners, thereby impeding beginners' contribution and learning. We conclude with a discussion that leverages the theoretical framework of knowledge building communities (Scardamalia, 2002), describing the potential opportunities and trade-offs of introducing competitive mechanisms to open knowledge building systems. We offer design implications for how developers might integrate competitive elements into systems designed for a community to facilitate public contribution.

4.2 Related Work

Our study builds on previous research on knowledge building communities, as well as prior studies on gamification and online open competitions.

4.2.1 Knowledge Building Communities: Benefits and Challenges

Knowledge building, stemming from learning science theories, is "the creation, testing, and improvement of concept artifacts" shared publicly in a community (Scardamalia, 2002). The building of shared knowledge has been studied as a prevalent phenomenon in open online communities, such as collaborative problem-solving in social Q&A sites (Tausczik et al., 2014b), co-creation of artifacts or wikis in open source projects (von Hippel and von Krogh, 2003; Fiesler et al., 2017c) and feedback exchanges in creative communities (Marlow and Dabbish, 2014a; Xu and Bailey, 2012b).

Knowledge building is considered to be critically beneficial to the health and growth of online communities, because it results in the expansion of the body of shared content, thus attracting new individual members (Von Krogh and Von Hippel, 2006; Kraut and Resnick, 2012). Scardamalia et al. introduces a framework of principles that mark successful knowledge building in a community: authenticity of problem, improvable ideas, idea diversity, abstraction, epistemic agency, collective responsibility, democratization of knowledge, symmetric advancement of knowledge, pervasive knowledge building, constructive use of authoritative sources, knowledge-building discourse, and transformative assessment (Scardamalia, 2002). These principles are realized in a variety of knowledge building online communities: in online communities focused on creative content generation, members add their creations to a collection shared by the community, thus offering others potential inspiration (Yu and Nickerson, 2011b) and the opportunity to build on their work (Dasgupta et al., 2016a). Community discussions about and feedback on such creations benefit the creators themselves in terms of giving them encouragement (Fiesler et al., 2017c; Campbell et al., 2016a) and diverse helpful/critical perspectives (Marlow and Dabbish, 2014a; Yen et al., 2016b). Additionally, such shared artifact-focused discussions benefit the community by interactively deepening their collective understanding of the creations (Kou and Gray, 2017b). Studies on social Q&A also show that online discussions can lead to collaborative problem-solving (Tausczik et al., 2014b; Li et al., 2015b), which generates new knowledge and promotes critical-thinking and innovation within the crowd (Tausczik et al., 2014b).

However, despite these benefits, maintaining regular, widespread public contribution is challenging because of the large scale and open-ended nature of online communities. Public knowledge building is based on community members' willingness to publicly present their creations and offer intellectual input to others (von Hippel and von Krogh, 2003). In most online communities, contributions are mainly made by a small group of individuals, leaving most other members as only occasional contributors or sometimes merely "free-riders" (Nielsen, 2006). Such low rates of contribution are a loss for the dynamic and diverse knowledge in the community. One reason that contribution rates are low is that it is difficult to constantly motivate participants to invest effort to build in addition to consume public goods (Benkler, 2002). Furthermore, contributors often have psychological barriers when considering whether to expose themselves to the whole community (Marlow and Dabbish, 2014a; Foong et al., 2017), especially when they are not that confident about the authenticity and maturity of their contribution (Kim et al., 2017b), even as such work may indeed

benefit the community. In addition, knowledge contributed to online discussions is often considered to be low quality (Agichtein et al., 2008b) and sometimes does not meet the community's expectations in terms of timeliness, investment, and substantiveness (Xu and Bailey, 2012b). Recognizing these challenges, we thus seek new ways to enhance knowledge building in open online systems.

4.2.2 Gamification and Open Innovation Contests

To encourage contributions to shared knowledge, many communities, including open innovation contest platforms, adopt gamification mechanisms. Such mechanisms include the use of extrinsic rewards (e.g., medals and rankings) as game elements for people to compete for in a non-gaming context. In some open online knowledge collaboration systems these mechanisms are prevalent (e.g., Stackoverflow) (Deterding et al., 2011; Bunchball, 2010). Extrinsic rewards are designed to facilitate overall engagement and commitment (Cavusoglu et al., 2015b; Bista et al., 2012), because they address a type of social need for some community members to increase their self-determination and self-efficacy (Richter et al., 2015). However, studies show that such extrinsic rewards can sometimes undermine engagement by suppressing community members' self-interest in the task itself (Bielik, 2012). In addition, rewards may result in less helping activities in the community, as members may be concerned about negative impacts on their rankings incurred by helping others (Zagalsky et al., 2016).

Open innovation contests, as a special type of gamified design (Moldovanu and Sela, 2001), have become an increasingly popular model for organizations to collect innovative solutions to open-ended problems from crowds (Leimeister et al., 2008). By adopting traditional gamification features such as medals, user achievement rankings and tangible rewards (e.g., monetary prizes), such contests introduce a competitive dimension to the interactions, wherein only one or a small number of solutions are selected as winners. Such features are implemented with the hope of motivating more solutions to be submitted (Morgan and Wang, 2010). Prior studies show that tangible rewards are the main reason that people are drawn to participate in such competitions (Antikainen and Vaataja, 2010), and that the bigger the prize, the more likely that there will be an increased number of participants (Zagalsky et al., 2016). In addition, people participate because of reputation rewards in the form of social attention from other participants, as the result of public rankings (Huberman et al., 2009; Antikainen and Vaataja, 2010).

While previous work has shown that crowd workers on non-competitive platforms would share advice and work opportunities in their communities to provide support to each other (Gray et al., 2016), less is known about how and why public knowledge would be shared on competitive platforms. While gamification elements (e.g., leaderboards) could increase engagement, it remains ambiguous to what extent competitive mechanisms enhance or suppress contribution to community knowledge, especially because public contributions that benefit the community may diminish one's chance of winning rewards.

4.2.3 Community Knowledge Building in Open Innovation Contests

In open innovation contests, competitors interact with each other and jointly discuss their innovations, but at the same time, try to individually contribute the best solution (Bullinger et al., 2010). A handful of studies have identified community knowledge building activities in open innovation contests. Some argue that competition can provide participants with common ground through which they teach each other domain knowledge related to the subject of competition, engaging those with less experience (Nag et al., 2012). Participants ask questions, evaluate ideas, and share experiences and information in public discussions (Hutter et al., 2011) where mutual commenting leads to more diverse solutions (Bayus, 2013). In competitions that allow exchanges of in-progress work, participants are able to revise and improve their own solutions by comparing their ideas with others' (LaToza et al., 2015). In competitions that allow people to form teams, participants contribute to broad discussions outside their immediate teams because they want to learn from different competitors (Bullinger et al., 2010; Füller et al., 2011); the better a team performs, the more they would share with the community (Zhou et al., 2017). On the other hand, competitive mechanisms may also result in less knowledge building activities, because only the winner will be recognized in the end (Lu et al., 2014). Participants may lose interest in learning about unselected solutions in the community and thus place less value on participation in public discussions (Chawla et al., 2012). Research has also demonstrated that participation in public discussions dramatically decreases after teams become stably formed (McInnis et al., 2018) and that many participants tend to freely take advantage of the ideas and artifacts shared by others without adding their own contributions (Huberman et al., 2009). A prior study of Kaggle Competitions shows that on average only a small portion of users share in-progress solutions in competitions, mostly when they are in an adverse situation such as not having enough time or teammates (Tausczik and Wang,

2017b).

Despite all these ongoing conversations about community knowledge building activities in open innovation contests, little is known about the competitors' motivations for contributing to public knowledge. It is unclear whether and how they balance competing and contributing, and whether participants across experience levels work with the same motivations. We thus investigated these open questions through in-depth interviews of active community participants. By understanding their reasons for contributing, as well as the challenges they encounter in this process, we hope to gain insights that may help guide the implementation of competitive designs to support shared knowledge building.

4.3 Empirical Setting: Kaggle Competitions

In this chapter we focus on Kaggle Competitions,[2] an important section of Kaggle.com. Kaggle is the world's largest data science online community, with 128,929 registered users from 194 countries at the time of this study. Kaggle has hosted 370 Competitions, sponsored by external organizations and companies seeking crowd-sourced solutions to real world data science challenges. Kaggle Competitions cover a variety of domains such as medical informatics, business intelligence, urban planning, etc., with a focus on prediction tasks, asking participants to compete for prediction accuracy.

We chose Kaggle Competitions as our empirical setting because, apart from its well-established competitive mechanisms (explained in 4.3.1), it affords community knowledge building activities such as public code sharing and social Q&A-based discussions. Public code sharing through Notebooks allows participants to directly share, replicate and build on each others' solutions (Tausczik and Wang, 2017b). It is a unique feature in Kaggle Competitions that is not on other open contest platforms such as TopCoder[3] and OpenIdeo.[4] In addition, Kaggle Competitions attract users with diverse background and experience levels (externalized by its user ranking system, explained in 4.3.1), allowing us to investigate the effect of competitive mechanisms on both expert and beginner participants. Finally, the diversity of topics covered by Kaggle Competitions also enables us to generalize our findings to different contest domains.

While in addition to Kaggle Competitions, there are many other features in the Kaggle eco-system,

[2] `https://www.kaggle.com/competitions` (permalink: `https://perma.cc/XV2A-L3M5`)
[3] `https://www.topcoder.com/challenges` (permalink: `https://perma.cc/VKA2-BRX9`)
[4] `https://www.openideo.com/` (permalink: `https://perma.cc/7WYS-CG3L`)

including user uploaded datasets, social news feed and online courses, in this chapter we focused only on Competitions and related knowledge building features — Notebooks and Discussion.

4.3.1 Competitive Mechanisms

Prizes and Medals

At the time of this study, 307 out of 370 Kaggle Competitions offered tangible rewards (271 with money, 22 with swag, 14 with jobs) as prizes. In each competition, participants can submit their solutions multiple times as individuals or in self-formed teams. After submission, participants immediately receive a score for their prediction and a rank of their solutions among all the others in the same competition, as shown in Figure 4.1. In most cases, only the top three ranked submissions are awarded prizes. Besides monetary rewards, Kaggle Competitions also features awarded medals (gold, silver and bronze) based on performance in a given competition. The specific rules on how a medal will be granted can be found in Kaggle documentation on its user progression system.[5] Medals show up on a user's profile page as an indicator of user achievement status, and are also counted towards the global user ranking system.

User Ranking

All participants in Kaggle Competitions are publicly ranked according to their cumulative performance, presented on its global leaderboard.[6] The user ranking system consists of five ranks (from the lowest to the highest): Novice, Contributor, Expert, Master and Grandmasters. The Novices rank is granted to users when they register. A user achieves the Contributor rank when they have their first submission to a competition. The ranks of Expert, Master and Grandmaster are based on the number of total medals a user gained.[5] Higher ranks in Competitions are more difficult to achieve. At the time of this study, there were 5,153 out of 128,929 (top 4.0%) users that achieved the Experts levels in Competitions, 1367 (top 1.1%) achieved Master level and only 171 achieved (top 0.13%) the Grandmaster level.

4.3.2 Community Knowledge Building Affordances

[5] `https://www.kaggle.com/progression` (permalink: `https://perma.cc/6BX7-GGTX`)
[6] `https://www.kaggle.com/rankings` (permalink: `https://perma.cc/4D47-6WSR`)

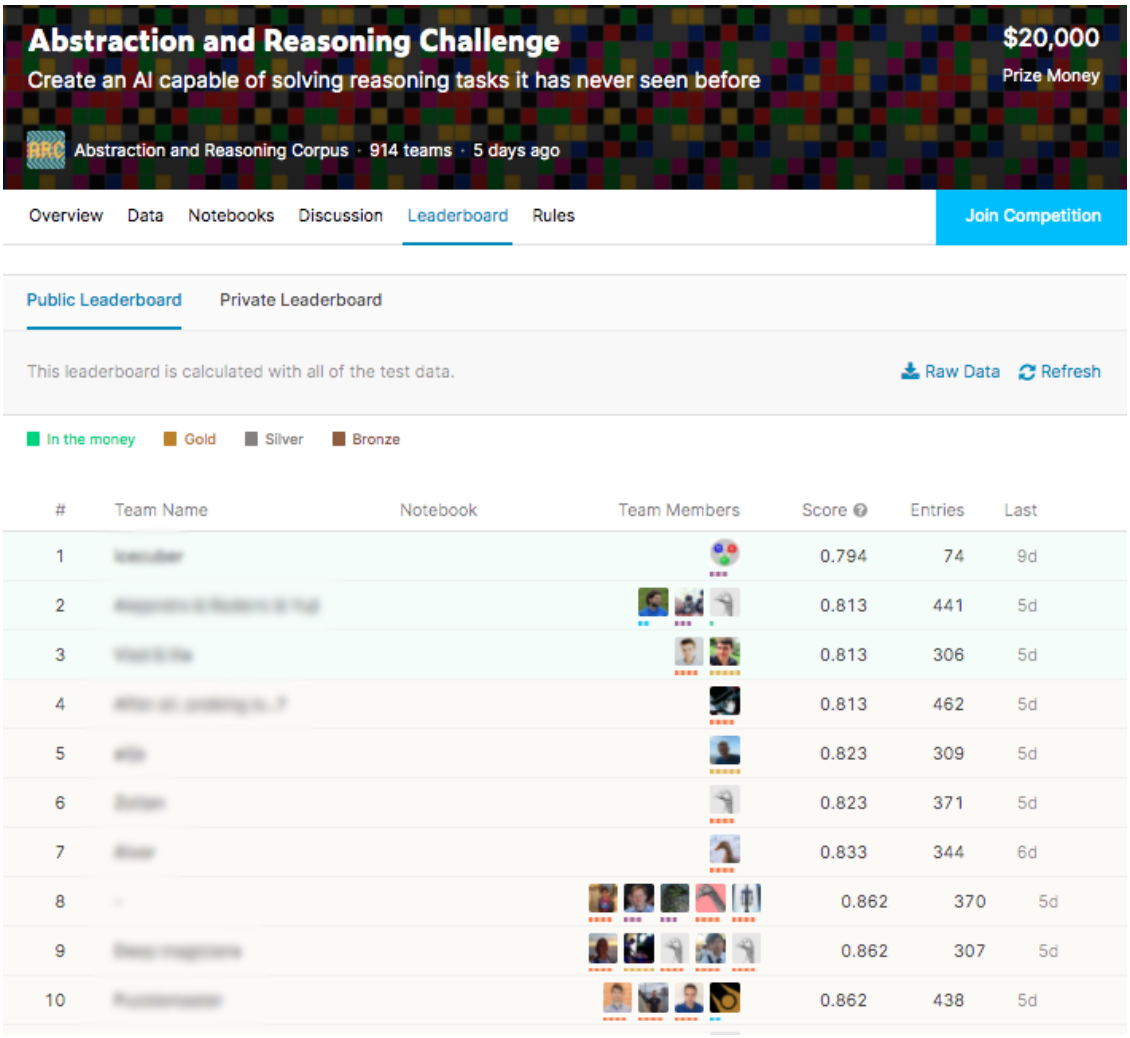


Figure 4.1: A snapshot of a leaderboard in a completed competition. The winners and medal receivers, along with the names of all team/individual participants, the scores of their submissions and their ranks in the competition are displayed publicly. Only top 10 rankers are included in this snapshot.

There are two major community knowledge building affordances in Kaggle Competitions: Notebooks for public code sharing and Discussion for social Q&A and text-based exchanges.

Notebooks

Notebooks[7] is a feature that allows users to share and execute others' code in a Jupyter Notebook environment embedded in and run by Kaggle. Users are allowed to modify others' shared notebooks and submit them as their own solution to a competition. At the time of this study, there were in total 11,658 notebooks shared on Kaggle, and as shown in a previous study (Tausczik and Wang, 2017b), around 10% of users have experience sharing notebooks. We need to note that not all shared notebooks are attached to a specific competition — some notebooks are shared with an educational purpose about general data science methods. Due to the purpose of this study, we specifically asked about usage of notebooks that are connected to competitions during our interviews. Similar to Competitions, the Notebooks section also contains a gamified component in the form of medals and has its own ranking system. Medals are awarded to a notebook based on the level of community approval in the form of upvotes.[8]

Discussion

Competition discussions are the forums attached to specific competitions, where signed-in users can post text-based ideas, questions and solutions as discussion topics and reply to others' topics. In addition to Competition discussions, there is also a general discussion forum that is not connected to competitions. As we study community knowledge building behaviors under a competitive structure, in this study we refer to Discussions as those attached to specific competitions. Users can also earn Discussion medals for their discussion topics and their contributions to someone else's discussion topics. Discussion medals are offered according to the "net votes," which are the sum total of upvotes minus the sum total of downvotes to a topic or comment. Discussion has its own ranking system as well.[8]

4.4 Method

4.4.1 Participants

[7] Notebooks were called "Kernels". In the middle of our study, Kaggle changed its name to "Notebooks". In this chapter we decided to call it "Notebooks" in order to stay consistent with the current configuration of the system. Some of our interview participants still referred to it as "Kernels" in the quotes presented in the following sections. We added a following "(notebook)" to "kernels" in the quotes.

[8] `https://www.kaggle.com/competitions` (permalink: `https://perma.cc/XV2A-L3M5`)

We conducted semi-structured interviews with 14 users to investigate their motivation and experiences when consuming and contributing to public knowledge in Kaggle Competitions. We posted our recruitment messages and a screening survey on the general discussion forum on Kaggle, related subreddits (e.g., r/kaggle, r/datascience) and slack channels for Kaggle users. We also sent individual messages to users on the user ranking leaderboard and the first author's social network. We recruited participants who have experience in participating in at least one competition (no matter if they successfully submitted a solution). We also purposefully sampled a pool of users from a variety of geographical regions to make our insights more generalizable. Each participant was compensated $10 for participating in the study.

We deliberately recruited both experienced and less experienced participants, as reflected by their professions and ranks in Competition, Notebooks and Discussion. In our analysis, we classified participants into two categories regarding their experience with data science: *experts* and *beginners*. Our definition of "experts" refers to participants who are self-identified as data science professionals AND have achieved Expert, Master or Grandmaster rank (top 4%, as explained in 4.3.1) in any of Competition, Notebooks or Discussion. We believe the combination of a career in data science and a top global rank is an indicator of advanced expertise in the domain. A total of 5 participants in our study fall into the category of experts. Compared to the world's top achievers, the other 9 participants are thus referred as "beginners." We chose this binary way to classify participants because we hope to see how competitive mechanisms affect experts and those with less experience in the same or different ways. We are aware that within the bracket of "beginners," there could still be nuances in levels of experience and expertise, which we did not consider in this analysis. We will address this point in the limitations section. After interviewing 14 participants, we were able to reach a saturation in the insights that we heard from them. In Table 4.1 we present the characteristics of our participants, including their gender, profession, region of residence, ranks in Kaggle Competition, Notebook and Discussion, and classification in our analysis (expert or beginner).

4.4.2 Procedure

Each interview lasted 30-60 minutes and was conducted via an online conference call. Interview questions focused on participants' experiences with competitions, code sharing, and discussions. To ground the interview, we first asked participants to describe and provide the rationale behind their participation in one

ID	Classification	Gender	Profession	Competition	Notebook	Discussion	Region
B1	beginner	M	Student	Contributor	Contributor	Contributor	Europe
E1	expert	M	DS professional	Expert	Contributor	Contributor	Europe
B2	beginner	M	Student	NA	NA	NA	North America
B3	beginner	M	NDS professional	Novice	Novice	Novice	North America
E2	expert	M	DS professional	Grandmaster	Contributor	Expert	Asia
E3	expert	M	DS professional	Grandmaster	Contributor	Expert	Europe
B4	beginner	M	NDS professional	Contributor	Contributor	Contributor	Asia
B5	beginner	M	Student	Contributor	Contributor	Contributor	Asia
B6	beginner	M	Student	Contributor	Contributor	Contributor	North America
B7	beginner	M	DS professional	NA	NA	NA	Asia
B8	beginner	F	Student	Novice	Novice	Novice	North America
E4	expert	M	DS professional	Contributor	Master	Expert	Asia
E5	expert	M	DS professional	Master	Contributor	Grandmaster	Asia
B9	beginner	M	NDS professional	Contributor	Contributor	Contributor	Asia

Table 4.1: Characteristics of study participants. In the columns of Competition, Notebook and Discussion rank, B2 and B7 chose to not disclose their Kaggle profile, so "NA"s are presented. In the column of Profession, "DS professional" refers to data science professional and "NDS professional" refers to professionals who are not working on data science related jobs. The Classification column shows the experience level defined in our study (expert or beginner). Despite our effort to include diverse participants in our study, we were only able to recruit one participant who are self-identified as female, while the rest all self-identified as males.

competition and then to share any knowledge building around that competition. We then asked them to reflect on their general practice and motivations more broadly when competing and contributing and using others' contributions on the platform.

The first author transcribed the interviews and then followed a thematic analysis procedure (Guest et al., 2011) to identify common themes across the interviews. As the themes were developed, they were discussed by the research team and checked for fidelity to coded samples selected from the dataset.

4.5 Results

4.5.1 Competition Incentivize Knowledge Consumption

Competitiveness is an integral part of the Kaggle experience. Our study participants, however, acknowledged that it is very difficult to actually win prize money in Kaggle Competitions even for experts, let alone for beginners: "even for Grandmasters, it is still very very hard to win the prize money," said E2, a Grandmaster level participant. As E2 explained, "the ratio of [time and effort] investment to [prize money] gain is so low." Although almost nobody regards the prize money as the primary incentive for participating in

Kaggle Competitions, participants, especially experienced data scientists, still consider competing for prize money, medals and higher rankings to be an important part of the experience. Expert participants regard the prizes as a concrete goal with which they are trying to advance their expertise: "for some of the competitions which I have a chance of winning, it's about trying to get the best result." (E5) Indirect financial benefits, such as new career opportunities, also drive participants to strive for a high ranking:

> "I worked as a consultant and most of my jobs come through Kaggle because people have seen my results in Kaggle. And so they offered me work. So I need[ed] to maintain my ranking." (E3)

The competitive atmosphere in the community, externalized by directly comparable scores and publicly visible rankings, motivates participants to improve their solutions so that they will compare favorably with others building knowledge about the challenge:

> "The way you do competitions is very different from the way you do your [data scientist] job. In competitions you will pay more attention to details, because you get a score after each submission; you get the ranks. In your job there is not a leaderboard, and you will not be directly compar[ed] with others, and you would think maybe you did okay, but actually you did not." (E2)

In general, participants regard the community as a healthy competitive platform and a resource for learning. Expert participants seek out opportunities to win, to build upon knowledge created by others, and to advance that shared knowledge. Beginner participants learn from the contributions of experts and enhance their own skills. In the following sections we elaborate on how experts and beginners leverage community knowledge differently.

Experts Seek for Diversity among Shared Knowledge.

During a competition, expert participants tend to read through numerous different notebooks in order to explore different ideas shared by the community. Our study participants described how notebook exploring behavior is especially common in their initial data exploration stage when joining a competition. We found that they like to understand the data thoroughly before diving into an analysis. They believe that building such an understanding helps them choose more suitable methods and models. In particular, they usually search for a variety of notebooks that contain code about data pre-processing and visualization methods:

> "I usually end up with going through the list, and opening a new tab 20 different times. And then I just work with

> the kernels (notebooks) and write down all of the stuff that looks interesting and then try to incorporate into my own script." (E1)

Expert participants, even those who are Grandmasters in Kaggle Competitions, believe they can learn from reading beginners' notebooks. Because beginners have less experience with competitions, they are less likely to be constrained by conventional ways of looking at the data: "newcomers are not here to win; they are here to try, so they really tr[y] out new methods" (E5). Experts regard beginners' notebooks as creative angles for understanding data, which might be used to discover novel approaches in future analysis. Another expert participant, E2, also shared his strategy of taking advantage of ideas shared in the discussions:

> "In my team, I'm usually responsible for scrutinizing every relevant discussion post that I could find, even those by beginners. Sometimes they actually post a question, but I can see the useful insights hidden in it" (E2).

Notably, experts could benefit from community knowledge contributed by beginners. They discover value and opportunities for innovation from beginners' contributions — sometimes not even realized by the beginners themselves. Therefore, beginners could be encouraged to contribute discussion posts or notebooks, leading to the advancement of ideas in the community.

Beginners Feel Empowered Using Experts' Contributions to Get Started.

Beginners, on the other hand, primarily use community knowledge generated by experts as an efficient way to enter into a competition. Code for data analysis and machine learning tasks can be very complex, typically including various stages and modules. Beginners take advantage of existing code so that they don't have to begin from scratch and then incorporate the code from those pre-existing notebooks into their own solutions. Beginners thus feel empowered when using notebooks that are highly upvoted and written by experts to get on-board:

> "You don't learn to build a bridge by inventing the bridge all over again. You go through the methods of bridge building from the experts that have come before you and then eventually innovate it in your own right." (B3)

Starting off from experienced competitor's notebooks can also give beginners a sense of how a good solution performs, or in many cases, understand what accuracy looks like compared with their own results: "primarily I use the [existing] notebooks just to know what is the average good score" (B7). Borrowing from

experts' notebooks also helps beginners start with a relatively high rank in the competition and recognize potential directions that could lead to improved results: "[high performance] notebooks can help you start with a very good leaderboard score early in the competition. So you just narrowed [the solutions] down" (B7). Therefore, beginners appreciated notebooks posted by experts that are directly usable as complete solutions. Those solutions as good learning resources are guiding the community towards solving the problems:

> "Kaggle is a competitive platform, but more importantly in my mind, it's an educational platform that the few that competes on a competitive level supports the masses... Whether they win or lose, they're going to have a good kernel (notebook) on their hands that they can post... Someone has that first strong kernel, followed by the second strong kernel, by the third strong kernel and [then] we integrate them." (B3)

While beginners' learning could be scaffolded by experts' solutions, one concern is that using limited notebooks written be a few experts may sometimes suppress creativity in the community. Because notebooks are easy to run and use, beginners may simply take in code that apparently generates good results without knowing how and why it works. This form of uncritical borrowing sometimes means that no further modification or additions to notebooks with good performance are made, as observed by an expert participant:

> "There's sort of group[s] where everybody's copying each other. So all the notebooks are different variations on the same idea after a while... I feel like it's so easy for somebody who's entering a competition just to copy somebody else's code. Not really doing anything innovative or understand[ing] what the code is actually doing." (E3)

From an individual's perspective, a bias introduced by using common solutions in existing notebooks sometimes diminishes their motivation for coming up with novel ways to approach the challenge:

> "If you're starting [by] looking at a lot of other people, you might get biased. And if you see that everyone is using some particular techniques, basically you stop thinking about how you can approach a problem." (B6)

As a result, innovation within the broad community can be diminished, ultimately generating fewer learning resources. Unexpected, creative ways of exploring problems, however, are what competitors, including experts look for in notebooks because often they are what pushes the overall competitive performance to a higher level.

4.5.2 Experts' Contribution to Community Knowledge: Motivations and Limitations

In this section, we describe our findings on how and why expert participants in Kaggle Competitions engage in activities around community knowledge building:

Experts Build Reputation through Public Sharing.

We found that in some cases, experienced data scientists contribute to notebooks and discussions for altruistic reasons. For example, E4 shared that his major motivation to spend extra effort to write clean and organized notebooks is that he would like beginners to benefit from his shared ideas. Nevertheless, we discovered that the major reason experts share knowledge to the community is to boost their reputations as data scientists, both on the platform as well as in the real world.

High reputation can potentially further lead to collaboration and network-building with experienced people in the community. While it is hard to win a competition as an individual, teams formed by experienced data scientists in general have more chances of actually winning the competition:

> "Once you achieve a certain status on Kaggle, it becomes easier to win other competitions because it's easier to form teams with talented people when you have a reputation." (E3)

Notably, in our interviews, we found that only expert participants had the experience of finding online teammates from Kaggle. Beginner participants, in contrast, did not find new people to collaborate with on the platform. The ranking system externalizes and quantifies skill level and expertise in a way that everyone in the community can see. Consequently, high achievers find it relatively easy to identify collaborators for teams, as their rankings serve as clear indicators of what they offer that would be mutually beneficial in a collaborative relationship:

> "When you get higher ranking in this platform, you get to interact with more highly ranked competitor[s], that gives you more opportunities for feedback and learning. So your ranking is your name card–like if I am ranked at top 10, I get to know the top 10 Grandmasters, versus if I were ranked at 1000th, then the top 10 people won't bother to talk to you." (E2)

The expert participants tend to find competitors who are at similarly high ranking, and who can also bring in different domain knowledge and techniques, in order to achieve higher scores. In a community of strangers from all over the world, public contributions are important ways to signal personal characteristics such as skills and domain expertise. E3 shared his experience learning about potential collaborators through their public contributions:

> "People write a lot on the discussion forums and they sort of drop clues about what kind of a solution they have. I just know, for example, there are people who are neural network experienced data scientists, [they say in discussion] 'I'm having trouble because I need more GPUs. So if they say that, then I know that they're using neural nets." (E3)

Notebooks and discussion posts could help experts get attention from other experienced data scientists who might become future collaborators. Therefore they share their contributions to communicate their status and expertise.

In addition to on-site reputation building, contributions to the community sometimes also lead to reputation building outside Kaggle. As Kaggle competitions are well-known as highly competitive and contain difficult data science problems, performance and reputation in Kaggle are linked to recognition in real life. Expert participants therefore contributed to notebooks in order to build up their real life data science portfolio:

> "I knew that I wasn't able to win this competition, but when I just posted a kernel that I already had, I could also improve my Google ranking. Like when people just look for my name, I might come up and they can say, hey, this guy knows about this or that topic." (E1)

Since Kaggle has a sophisticated scoring and evaluation system, a solid performance record on Kaggle is considered by some companies as an indicator of real world data science skills. Putting thought and effort into organizing an exceptional notebook is considered by experts to be an easier task than winning a competition, but one that offers publicly viewable skill sets can have a positive impact on the contributor's career development, leading to indirect financial benefits. Experts, who are more likely to receive good scores than beginners, are thus motivated to publicly share their notebooks.

Experts Primarily Share Abstract Solutions

While some experts share their solutions to competitions in well-explained notebooks and detailed discussion posts, we discovered a pattern from our expert participants that they tend to only share "abstract" solutions. Abstract solutions — usually in the forms of a notebook with snippets of code stripped from a full solution, or a text-based high-level description of methods in discussion posts — can be used to communicate ideas but not replicatable results. One reason that experts only share partial solutions is that in order to

prevent ranking inflation that would result from many people simply running a high scored notebook, Kaggle moderation team urges contributors to remove notebooks or discussion posts containing highly scored full solutions posted before a competition ends: "[Kaggle] let people use kernel (notebook) to communicate ideas, on an inspiration level, but not submittable solutions" (E2).

Apart from being discouraged by the platform, posting full solutions publicly can also result in complaints from the community and potentially losing reputation. E5, a Master level participant in our interviews, recalled his experience posting detailed documentation of his highly scored solution in a competition discussion. He received down votes and protest comments from competition participants who had some experience themselves, as their rankings were negatively impacted by his sharing:

> "Who really care[s] is the middle range [competitors], because what you do affect[s] their results... So these are the people that don't like people [to] reveal their secret[s]." (E5)

As a result, experts avoid sharing their solutions during a competition. Instead, in cases where they do contribute to notebooks and the discussion, they share only snippets of code or partial solutions described in an abstract way. Those incomplete shared ideas are viewed favorably by highly achieved participants, as they often already know the dataset and methods very well before they look at what is shared. Thoughtful participants can understand the problem that the abstract solutions try to tackle and know how to leverage them:

> "The snippets are better because they are focused into one topic… they are very important and informative. I don't want end-to-end solutions because it's kind of a copy paste, which I don't like." (E4)

While abstract solutions might be understandable by experts, beginners may experience such solutions differently: "for someone who did not fully invest in a competition, they probably won't have a deep understanding from the post." (E2) The educational value of incomplete solutions to the general community is therefore discounted — a point we will further discuss in section 4.5.3.

Experts Prioritize Competition over Contribution

While experts may not share full solutions during a competition, what about after a competition is over when sharing full results will not undermine the fairness of a competition? Will the advanced competitors organize their snippets and present them in a cohesive and comprehensive way? The answer is "no" from our investigation.

Making notebooks well-explained and organized is considered time-consuming. We found expert participants believe the scripts that they use for competitions are not clearly structured so that they are not suitable for public consumption. For example, E3 commented on the reason that he rarely posts his solutions as notebooks:

> "I don't share my code. Not because I think it's a bad idea or because I'm competitive, but I'm just embarrassed by my code... It's not very clean" (E3).

Further, some expert participants (e.g., E2), as data science professionals, write and run most of their programs on local machines instead of on notebooks which live on the Kaggle server. Therefore, sharing their full code as notebooks would require extra work to migrate code from their machine to Kaggle on top of all the effort put into competitions, which is not viewed favorably by busy professionals.

Since the cost of organizing code is high, and there are always competitions being launched in the community, expert participants prioritize budgeting their time for diving into the next competition:

> "There are a lot of competitions running concurrently. I mean I would rather spend my time in the next competition, rather than [a] competition that is already passed." (E5)

As abstract shared solutions are understandable by other experts and can potentially bring them reputation, and the next competition can potentially bring them more opportunities for winning, expert participants are not motivated to look back and assemble their partial results into a cohesive solution once a competition is over.

4.5.3 Beginners' Engagement with Community Knowledge Building: Challenges

Beginners Feel Vulnerable Exposing Their Newbie Status

While expert participants contribute notebooks and discussions that are appreciated by other experts in the community, we found that beginner participants seldom make public contributions. As achievement and experience level are clearly quantified and externalized in the community, beginners live with a newbie's identity — whenever they participate in public activities, their profile will indicate that they are less experienced. Because of this, beginners feel vulnerable when exposing their ideas, opinions or questions to the public, worrying that more experienced members will not take their contributions seriously:

> "If I were to have a question, I don't think I would have been comfortable posting. There's always some anxiety that, you know, everyone else is more experienced at this than you, and they're gonna think my question is stupid

> sort of thing. I think that was the impression that I got is that the people who were posting and asking questions really knew what they were doing." (B2)

Apart from sharing original knowledge and ideas, beginner participants also reported difficulty in joining in "expert niches" and adding to their knowledge building activities. For example, participants shared that it is hard to add to an ongoing discussion thread because discussions are usually driven by certain exclusive groups of experienced users: "it's more like a conversation… it's basically quite nested" (B4); "They seem to already know each other, [so] there is no point for me to cut in" (B9). They also perceive the discussion on Kaggle to be "more formal than Stackoverflow" (B5) and other programming or statistics help channels because Kaggle discussions include many highly visible, established people from the community. As a result, newcomers feel hesitant to contribute potentially superficial knowledge and indeed seldom post in general, driven away from presenting themselves to and getting input from the community.

Beginners Have Trouble Gaining Visibility

For beginners who do publicly share their contributions, they do so to communicate with other community members and to learn from their advice: "when I share the notebooks, I want other people to see my code then help me improve it" (B9). However, as the ranking system gives those high achievers more visibility in the community, notebooks or discussion posts written by beginners seldom gain attention and support:

> "We posted one or two of the good modeling step we made. Didn't really get any attention [be]cause like we were not that good or anything. But definitely I'd be happy if people could take a look at it and have questions about it." (B1)

We also learned that an important criterion for participants to determine whether a notebook is valuable is whether the author is a highly ranked competitor (B4, B5, B7 and B9). Therefore, beginners' solutions rarely receive views or encouragement. As reflected by an expert participant, it is rare to encounter shared solutions written by beginners:

> "There are so many novices who write excellent notebooks… there are so many people who are doing it well, but they're not able to show up (in rankings)." (E4)

Furthermore, because high achievers' notebooks are more visible in the community, beginners believe that only notebooks or discussions resulting in good performance are valuable to the community. Therefore, they feel discouraged from contributing:

> "The thing with notebooks on Kaggle for you to get feedback is that everything has to be properly and sophisticatedly written, like with headings and everything, in order for it to be run higher on the list and to gain visibility and then people actually apply it, which I was maybe lazy enough to not do that." (B6)

One result of the low visibility of beginners' contributions is that it is hard for them to build reputation in the community and thereby to find collaborators. While the ranking system is beneficial for high achievers to showcase their skills and find teammates, it becomes a barrier for lower rankers to find teammates on Kaggle as they cannot build up a portfolio with highly ranked solutions:

> "I just sent quite a few proposals, but people haven't accepted... My profiling is not that good, so people just stick with high ranking people" (B7).

As a result, beginners are separated from experts. Having no chance of working with experts means having no chance to directly learn from the top minds on the platform. Therefore, it is very difficult for someone who is new to data science with the purpose of learning to rapidly develop their ranking in the community.

Beginners Face Difficulty Using Abstract Solutions

Although beginner participants in general appreciate solutions that are publicly shared by experts, they are often confused by notebooks that only contain partial solutions, because they do not have the technical skills and knowledge to figure out all the dependencies and next steps independently:

> "It is common that they have some very unfriendly connections between codes. You have to design the modules yourself, and you need to add your own tool packages... Just sometimes, you can't see their whole pipeline [of code]." (B9)

Given the norm that experts mainly share notebooks that contain only a snippet of code or partial solution, it is very difficult to understand the notebook author's completed approach to the problem. Most experts' sharing thus does not help with either their problem solving or general learning — instead, such sharing creates barriers for beginners to utilize and offer insights to the notebooks.

Beginners' difficulty in comprehending experts' abstract solution may further drive them away from participating in the community. B8 shared a story about her first time participating in a competition: in the beginning she was motivated, so she looked up multiple highly scored notebooks and discussion posts, but could not get any of them to integrate with her own solution:

> "Some people only put a snippet, then sometimes I don't exactly understand what they did after that; and when I want to do the same thing, it just doesn't work." (B8)

This experience made her feel unconfident about her ability to solve the problem. As a result, she dropped out from the competition without submitting a solution. While beginners need a directly usable solution to get on-board and feel a bit of empowerment despite their already low public status, failed attempts with abstract solutions can undermine their motivation to get involved in the competition, let alone contribute their own ideas to the community.

4.6 Discussion

In this chapter, we presented our findings from 14 in-depth interviews with both experienced and less experienced participants in Kaggle Competition. We unpacked how participants consume and contribute to community knowledge under a competitive system. Notably, although Kaggle Competitions is viewed as a successful platform that leverages competitive incentives to curate a large amount of independent submissions, our investigation indicates that the current design of Kaggle Competitions does not equally support community knowledge building activities among both experts and beginners. Competitive mechanisms incentivize experts to engage in public knowledge building activities, but present challenges for novices. Experts, who leverage notebook and discussion posts to build reputation in their own niche, tend to share abstract solutions that are hardly usable by beginners and prioritize competing over organizing their contributions. Beginners, on the other hand, face anxiety and difficulties in getting involved, gaining attention and encouragement for their contributions, as well as in comprehending knowledge artifacts contributed by experts. The challenges for beginners to contribute to public knowledge collection echo the fact that despite Kaggle's large user base, only 10% of users have contributed a notebook for public usage (Tausczik and Wang, 2017b).

Our findings add to the ongoing discussion about competitive design in knowledge building systems by surfacing the different challenges and opportunities such designs introduce for expert and beginner participants. In this section, following the framework of Knowledge Building Communities (Scardamalia, 2002), we analyze our findings to explore how competitive mechanisms may positively or negatively impact knowledge building activities in an online system. We specifically focused on 4 principles identified

in Scardamalia's framework for characterizing a successful knowledge building community: symmetric knowledge advancement, democratization of knowledge, knowledge building discourse, and idea diversity. We specifically chose to discuss these 4 principles because we found these were the particular dimensions of knowledge building communities that could be impacted by the dynamics between experts and beginners based on our findings. We acknowledge that competitive mechanisms could be leveraged to motivate symmetric or mutually beneficial knowledge advancement particularly among experts, but not among beginners or between experts and beginners. We also surfaced resulting expert niches in the community, which could undermine democratization of knowledge and knowledge building discourses among beginners, impacting idea diversity in the community.

4.6.1 Symmetric Knowledge Advancement Among Experts

Competitive mechanisms enhance symmetric knowledge advancement among experts in the community. Symmetric knowledge advancement refers to equalized knowledge exchange in the community, where members both obtain knowledge from others and produce knowledge that others can use (Scardamalia, 2002). While it is not hard to understand that experts consume knowledge added by other experts in order to improve their performance in the competitions, our study unpacked why experts in turn contribute to public notebooks and discussions even though such public contributions could potentially undermine their opportunities for winning.

We found in our study that the high prize money in Kaggle Competitions, though extremely hard to win, attracts many experts in the field (who have at least the potential to win) to invest their effort. Only a small number of competitors can win a given competition, but that does not seem to dissuade experts from engaging in the challenges. In general, expert participants do not worry that their public contribution will undermine their already very small chances of winning. Though they recognize they are unlikely to win rewards, because of the highly competitive environment, scoring a high achievement in a well-known data science competition community is a sign of honor. Echoing the literature on gamified design (Bista et al., 2012; Cavusoglu et al., 2015b), our findings show that experts are motivated to earn medals and higher ranks in order to gain benefits both within the community and beyond. High reputation in the community offers a powerful extrinsic motivation (Kraut and Resnick, 2012) to join in the public contribution, as contribution

can lead to reputation boosting (Antikainen and Vaataja, 2010; Huberman et al., 2009). Our findings add that reputation under competitive structure leads to collaboration opportunities with other experts, aiding experts to advance even more in the competitions. The unique benefits of reputation introduced by competitive structure are effective incentives for experts to become both consumers and active contributors to public knowledge, achieving symmetric knowledge advancement.

4.6.2 Expert Niches Driving Away Beginners

The symmetric knowledge advancement observed among experts, however, does not generalize to the rest of the community. While we found that competitive structure incentivizes experts' knowledge building activities, we also recognize its negative impact on beginners' participation on such activities. This negative impact is primarily because the competitive mechanisms result in the formation and over-representation of expert niches in the community. Beginners are excluded from experts' knowledge because it is not comprehensible and usable for them. Further, in niches of expert participation, beginners feel vulnerable about exposing themselves.

Undemocratized Knowledge

Democratizing knowledge is an important principle in a healthy knowledge building community, according to Scardamalia's framework. The democratization of knowledge in a community requires all participants, regardless of their levels and background, to be able to contribute and consume knowledge. Our study shows that a competitive system design, however, prevents the equalized consumption of knowledge among beginners.

Under a competitive structure, it is easier for experts to form a niche with other experts, resulting in the produced knowledge being consumed only by those within the niche. First, while moderators in online competitions generally support participants and enforce competition rules (de Souza et al., 2020; Machado et al., 2019), we found that in order to maintain fairness in the ranking system, they would also intervene in the community's public knowledge sharing. For instance, Kaggle moderators would prevent participants from sharing detailed solutions — only ideational and abstract level sharing is allowed. Such abstract information does not impact experts' performance in the competitions, nor does it impact their advancement

in the community, since experts have the background to understand abstract solutions and it is easier for them to find collaboration teams. However, the policy against sharing detailed information creates challenges for beginners who come in to learn and have not developed the skills to comprehend and leverage abstract solutions. It leaves less scaffolding for them to reverse engineer good approaches thus undercutting their opportunity to improve their own rankings.

Second, although experts recognize the value of community learning, because of the competitive nature of the community and the potential benefits of high achievement in a competition (financial benefits and reputation), they tend to prioritize competing over activities that further knowledge building. For example, because it takes extra time and effort to organize their code and solution to make them detailed and comprehensible for the public, it is natural for experts to choose to move forward to another competition instead. This finding is supported by literature where extrinsic rewards can distract contributors from producing public goods, even if they would feel interested in doing so (Bielik, 2012).

Their contributions, consequently, are concentrated during the competitions for the purpose of interacting with other experts. Their perceived audience are other experts like them, who do not mind abstract or partial solutions or nested discussion conversations. As experts usually only collaborate with experts and therefore are largely disconnected from beginners, they usually do not fully recognize the beginners' desires to learn from their contributions.

Exclusive Knowledge Building Discourse

The niches of expert knowledge further result in patterns of expert only participation. In Scardamalia's framework, Knowledge Building Discourse is an important pathway to the sharing, refinement, and advancement of community knowledge. According to this principle, participants with all backgrounds and levels should participate in discussion and critique that leads to knowledge advances achieved by the group. While previous research recognizes that competitors participate in and benefit from idea and feedback exchanges (Hutter et al., 2011; Nag et al., 2012), we took a closer look at participants of different experience levels. We found that, on the contrary, the competitive mechanisms in Kaggle may undermine such discourse among beginners.

When experts create niches in which they share knowledge that can only be used by other experts, begin-

ners are excluded from effectively leveraging the information and joining the discussion and exchanges, an essential activity for collective knowledge sharing. This disjuncture feeds back to the community's activities, where beginners do not see a lot of other beginners in the discussion. As the ranking system externalizes and quantifies individual competitors' skill levels, it is clear at a glance who is experienced and who is a beginner. Therefore, beginners feel more pressure when they desire to share their solutions and ideas. This echos previous literature that in professional online communities with reputation systems, beginners feel apprehensive about contributing content because they feel stress about being perceived as a "rookie" by the experts (Marlow and Dabbish, 2014a). In a competition community, the mechanisms of prize money, medals and public ranking exacerbate the experience level hierarchy, leading beginners to experience more social pressure and reasons to doubt their worthiness in the community.

Another aspect of niches populated by experts is the distortion of collaboration. As indicated in literature, competing in teams leads to better solutions than joining a competition as individuals (Boudreau and Lakhani, 2015; Zhou et al., 2017). However, as experts form teams in their niches, beginners, who are buried in the leaderboard, and who do not actively participate in public knowledge building activities (e.g., discussions), become more distant from those in the expert niches. While the system provides experts with a convenient way to identify similarly ranked potential teammates from whom they are likely to learn new skills, beginners do not have a chance to directly collaborate and work with high ranking contributors. This differentiated experience can result in polarization of levels in the community; experts become more closely collaborative, while beginners struggle because they receive too little direct feedback on their work and lack the courage to ask for it.

4.6.3 Impact on Idea Diversity

One potential negative impact from the expert niches to the community is less inspiration. Idea Diversity, as stated in Scardamalia's framework, is an important dimension of a good knowledge building community: "idea diversity is essential to the development of knowledge advancement, just as biodiversity is essential to the success of an ecosystem."(Scardamalia, 2002). Echoing this principle, participants in our interviews expressed a desire for diverse contributions. We learned that expert participants read through a large number of notebooks, especially when first approaching a problem, because they want to be exposed to different

ideas. Notebooks from both beginners and experts, notebooks with high scores and low scores—all can potentially be valuable to all levels of participants in the community. Diversity, to some extent, stems from the variety of experience levels among contributors. Beginners are experience-less, so they are not afraid to try non-traditional but innovative methods. Experts can build upon and revise those innovative approaches to further the performance of their solution. However, a lack of beginners' contributions in the community can hinder experts' learning, and in the end further negatively impact the advancement of shared community knowledge. Because the experts cluster in niches, beginners are discouraged from sharing their bold ideas. Experts thus lose out on opportunities to be inspired by beginners, which impacts the overall knowledge advancement in the community.

4.7 Design Implications

4.7.1 Using Competitive Mechanisms to Motivate Experts' Contribution

We offer design implications for the developers of new online knowledge building systems. In general, the inclusion of competitive mechanisms will motivate experienced users to contribute to sites designed for knowledge building. For a crowd-sourcing system that lacks experienced input, prizes that are hard to achieve can be used as incentives. Such incentives can attract more experienced contributors, who can potentially contribute valuable knowledge, spurring more general participation among the community. To facilitate knowledge building behaviors such as idea sharing and feedback exchange, reputation-based incentives (e.g., rankings and medals for sharing behaviors) can also be embedded in the design.

4.7.2 Encouraging Experts to Produce Beginner-Friendly Contribution

On the other hand, designers of open knowledge building systems should also recognize the limitation of expert contributions resulting from competitive designs. As they are motivating experienced participants to provide input, designers should also consider ways to guide them to contribute knowledge that could benefit all levels of users in the system. As detailed sharing may harm the fairness of a competition, future systems could find ways to motivate experts to produce well-organized solutions after competitions are over. For example, a new reputation system could be designed to specifically acknowledge authors whose contributions

are favored by beginners. Further, future competition designs should emphasize the completeness and understandability of what is shared, and award such shared content with more tangible credits (e.g., monetary prizes) even after the competition is completed.

Apart from innovating on motivators, designers could also consider including more scaffolding mechanisms that can guide experts to generate complete and comprehensible knowledge artifacts. For example, a feedback system could be implemented into the input interface, highlighting potions of the knowledge artifact that needs more elaboration. Systems could also include rubrics and examples following principles such as cognitive apprenticeship (Collins et al., 1991), prompting experts to share accessible contributions.

4.7.3 Removing Barriers for Beginners' Engagement

In addition, designers need to pay more attention to beginners when introducing competitive mechanisms to any knowledge building system. Our study shows that compared to experts, beginners tend to feel apprehensive about contributing their ideas to the community, because ranking systems can externalize their newbie status. Future system design should be able to help beginners overcome such pressures. The design of future systems could explore new ways of presenting reputation status, perhaps through the choice of anonymity, so that beginners share their knowledge artifacts without the concern about their social image in the community. Also, systems could innovate on their reputation system, allowing beginners to gain special credits that can boost their status when contributing to community knowledge.

4.7.4 Increasing Beginner-Expert Interactions

Last, but not least, we also find that experienced participants and beginners are largely disconnected. To some extent, beginners are invisible to the experts as experts have their own closed connection networks. However, as we found, beginners use experts' contribution as scaffolding for starting to compete, and experts leverage beginners' sharing as a source of novel ideas. We thus suggest that system designers develop mechanisms to connect beginners with high status participants in the community. For example, matching mechanisms could be embedded in the team formation process so that beginners will have a greater chance of working directly with experts; new reputation systems, for instance, a "mentoring badge," could be established to motivate experts to team up with beginners and invest extra effort in guiding them. At the

same time, designers could also design for collaborative activities within teams formed by both beginners and experts, building on models of legitimate peripheral participation and situated learning (Lave et al., 1991). The teamwork should be designed so that the beginners can participate in experts' problem solving in a way that empowers them with some tasks that they are competent with, but also will not distract from experts' work.

4.8 Limitations and Future Work

While we believe that our study contributes several empirical insights about knowledge building communities and competitive designs, we admit that our study could be extended in a few ways. In this chapter, we mainly discussed how competitive mechanisms might impact two types of participants, the highly achieving (i.e., experts) and the less experienced (i.e., beginners). Due to our limited time and funding, we were not able to recruit a lager set of participants with more diversity in their experience (e.g., intermediate data scientists). We are aware that this dichotomous way of classifying participants may collapse some nuance in both categories — for example, within the bracket of experts or beginners, there might be differences in experience and expertise. Although we were able to discover many common patterns using this simple classification, this is a limitation.

In this study, we chose interviewing as our method because we would like to investigate participants' motivation, practices, and challenges in knowledge building under a competitive system in-depth. Due to the nature of our recruitment strategy, all participants were self-selected for this study, which may threat on the validity of our insights. In addition, while we were able to inductively identify many prominent themes, we did not study the prevalence of these themes, nor do we offer causal or statistical evidence on the relationship between participants' experience levels and their knowledge building activities. Future research could build on this work to carry out a quantitative analysis on the user profile and public contributions (e.g., notebooks, discussion posts) in Kaggle, testing if our qualitative insights hold in a larger scale.

Last, but not least, our study focuses on a single platform, Kaggle Competition. While Kaggle is the most prominent and frequently used online data science contest platform, other smaller examples exist and should be considered in future research. Our work is further constrained in that we closely invested how specific competitive design on Kaggle (prizes, medals and rankings) affect the usage of specific knowledge

building affordances (notebook and discussion forum). Future work should explore whether similar patterns exist in many of other competitive systems with different implementations of competitive and knowledge building features.

4.9 Summary

In this study, I discovered that while Kaggle members primarily aim to enhance their technical skills, the ability to recognize, reflect on, and effectively communicate the process of working with large datasets is crucial in informal contexts. This observation echoes the growing scholarly focus on process communication in data science. Researchers are increasingly paying attention to understanding how learners explain their rationales and the exploratory steps they take when engaging with data (Rule et al., 2018b). Recent design principles for data science learning platforms also emphasize introducing learners to the often messy process of creating and categorizing data in the midst of uncertainty and complexity (D'Ignazio, 2017). Furthermore, Muller et al. (2019) highlighted that a distinguishing trait of experienced data scientists is their ability to identify elements of the "creation" and "design" of data work, such as human assumptions in establishing ground truth.

In addition, this work reveals how people of different experience levels in data science collaborate on analyzing and problem-solving with large datasets. As shown by the specific case of Kaggle, such collaborative ability involves the identification and collaboration with community members with desired data skills. It is also crucial to tailor communication of the exploratory procedures to various audiences and purposes, including collaborating with immediate collaborators, exchanging abstract analysis and feedback during competition, and post-competition sharing to signal reputation and skill sets.

Among these practices, I identified a disconnect in the way beginners and experts engage in collaborative knowledge building within their competitive structure. While both groups value each other's input, the competitive features of Kaggle influence their behaviors differently. Experts often produce complex knowledge artifacts with the goal of enhancing their reputation within specialized areas. Although valuable, these artifacts can be too complex for novices to fully understand or use. In contrast, novices often feel vulnerable and overshadowed due to the platform's competitive nature, which may deter them from sharing their own insights and solutions. This study underscores the need for systems that emphasize the importance of

process sharing among different levels of expertise and promote more inclusive and democratized learning environments.

Chapter 5

Study C: Critical Data discourses on Twitter

Prologue

Study C took place between 2021 and 2022, with the majority of data collection and analysis conducted in 2021. Most recent research has underscored the importance of fostering critical data literacy through the public's engagement with data related contemporary social issues. However, a 2020 study on anti-masker communities on Twitter and Facebook during the COVID-19 pandemic presented a significant challenge. Lee et al. (2021) showed that conspiracy communities employ the same strategies promoted by critical data literacy advocates, including critically evaluating the context and ethics of data collection, processing, analysis, and decision-making, to further their narratives. This issue is intrinsic to informal settings due to their open discussion nature.

The concerns highlighted in Lee et al. (2021) underscore the challenges of promoting critical data literacy in informal online environments. Do online communities and social media platforms present a treacherous landscape for individuals engaging with data, given the potential pitfalls of conspiracy discourses? Is it possible to distinguish between practices of critical data literacy and conspiracist narratives? More broadly, what does critical data literacy encompass within these informal online settings? Motivated by these questions, and with a belief that individuals can be guided towards critical data literacy without falling for conspiracist narratives, I conducted on this study, focusing on online discussions surrounding COVID-19 vaccine data on Twitter.[1]

[1]The data collection in this study had been completed prior to major policy shift and the change of branding of Twitter—now

The discourse on vaccine data has long been shadowed by conspiracy-driven claims and rumors even before the COVID-19 pandemic. I collected a large dataset of public Twitter posts that contained pro- and anti-vaccine discourses related to COVID-19 vaccine data in the first six months following the introduction of the vaccines. Through a series of theory-driven statistical analyses, I found that certain textual features—such as assertiveness and causal assertions—are more likely to present in anti-vaccine discussions as compared to their pro-vaccine counterparts.

As of September 2023, when this dissertation chapter was written, the study is being prepared for submission to a general science journal. The following sections, from §5.1 to §5.4, showcase the unmodified manuscript that I am preparing. The manuscript was a collaborative effort between myself, Aaron Shaw, and Benjamin Mako Hill, in which I was the lead of the work and was responsible for the collection and preparation of the datasets, the design and conduct of the analyses, and the writing of the manuscript. Section §5.5 offers a brief summary, highlighting the key insights of this study in relation to the thesis of the dissertation.

5.1 Introduction

The public routinely engages with data about vaccines and vaccination campaigns in online social media (Gunaratne et al., 2019). Such widespread and often critical engagement with vaccine data in social media can have positive impacts on public health and society, but remains vulnerable to conspiracist claims and other threats to information integrity (Broniatowski et al., 2018; Casciotti et al., 2014; Kennedy et al., 2011; Warren and Wen, 2016). While experts and machine learning algorithms may be able to differentiate anti-vaccine discourses about data from more reliable and accurate information, non-expert people often cannot. Conspiracist discourses about public health issues in social media adopt many of the same rhetorical tactics as scientific discourses that use data as evidences (Lee et al., 2021), but do so in order to undermine trust in orthodox, credible information about reliable and safe public health interventions. Effective responses to these threats must include a variety of strategies, including the identification and dissemination of concrete ways to distinguish anti-scientific discourses about vaccine data. However, many social media users lack the sorts of domain expertise or statistical knowledge that would help differentiate reliable discourses. In

known as X—in late 2022 and 2023. Certain features of the platform mentioned in this study may no longer exist.

addition, indicators of professional standing, expertise, or scientific validity are often absent in social media environments, preventing the use of such signals to identify sources of credibility.

The pervasive misinformation and controversies around data about vaccines during the COVID-19 pandemic illustrate some of the challenges involved. In social media, millions of people observed or participated in discussions about data related to the COVID-19 vaccines, shaping their decision-making and perceptions regarding public and personal health (Lyu et al., 2021). Numerous scientists and public health officials provided consistent and reliable information regarding data about the safety and efficacy of COVID-19 vaccines before and during their deployment around the world, but such efforts met with skepticism and outright resistance (Malova, 2021; Kumar et al., 2022), leading to lagging vaccine uptake, preventable deaths, and disparate impacts of the pandemic (Cuadros et al., 2022; Martin et al., 2021). These experiences have historical precedents, as well as further ramifications as vaccine hesitancy and resistance have expanded as the COVID-19 pandemic recedes (Ashton, 2021).

To help address these threats, our study contributes a novel approach to distinguishing online anti-vaccine discourses about data, opening a pathway to promoting the public's critical engagement with vaccine data in social media. We propose and evaluate textual attributes, identified in prior work as characteristic of conspiracist discourses, that distinguish anti-vaccine social media discussions of data.

Specifically, we examine online discourse about data related to COVID-19 vaccines on Twitter (which we refer to as *data discourses*), and we pay special attention to two types of discourse that leverage data to substantiate their claims: those that support the distribution and use of COVID-19 vaccines (referred to as *pro-vaccine*), and those that illustrate vaccine skepticism (referred to as *anti-vaccine*). The COVID-19 pandemic illustrates the urgency of vaccine skepticism, as it led to excessive COVID-19 infections and deaths (Cuadros et al., 2022; Martin et al., 2021). Such skepticism towards the COVID-19 vaccines, ranging from direct denial to delay in acceptance of vaccines (MacDonald, 2015), has become a global issue (Cuadros et al., 2022; Almaghaslah et al., 2021; Lockyer et al., 2021; Schernhammer et al., 2022; Wang et al., 2021; Okubo et al., 2021). For many countries, the availability of COVID-19 vaccines has no longer been a concern since mid-2021, yet the skepticism towards vaccines poses a major obstacle to vaccination at scale (Aw et al., 2021). Beyond the COVID-19 pandemic, vaccine skepticism has been a threat to public health for a long time (de Figueiredo et al., 2020) and in 2019, the World Health Organization (WHO) listed vaccine

hesitancy among the top 10 threats to global health.

Among the many factors that lead to vaccine skepticism in the general public, online anti-vaccine discourse plays a significant role (MacDonald, 2015). During the COVID-19 pandemic, misinformation spread widely via social media (Goel and Gupta, 2020). Scholars coined the term "COVID-19 infodemic" to describe how (mis)information about the global pandemic from questionable sources also spread like infectious diseases on multiple social media platforms (Cinelli et al., 2020). Among all kinds of information, opinions toward vaccines are among the most frequently discussed content about COVID-19 on large social media platforms such as Twitter (Lyu et al., 2021). Anti-vaccine campaigns have adopted large social media platforms such as Facebook to broadcast messages to vulnerable groups who are unsure about vaccines, and a single piece of anti-vaccine content can easily reach an audience of tens of thousands of people in a few days (Jamison et al., 2020). These discussions contain numerous expressions of vaccine hesitancy, such as talk about potential side effects and safety concerns (Malova, 2021; Kumar et al., 2022), and such claims began to spread while the vaccines were still in trials (Megget, 2020). Many anti-vaccine discourses incorporated conspiracy theories, including claims that the COVID-19 vaccines would inject microchips into human bodies as a monitoring method by the government or that mRNA-based vaccines would result in "Genetically Modified Humans" (Hotez et al., 2021). Such misleading arguments can lower confidence in the vaccine and exacerbate vaccine hesitancy (Loomba et al., 2021).

The scientific community has explored ways to encourage critical public engagement with data as a step towards mitigating anti-scientific and anti-vaccine attitudes. Promising examples include citizen science platforms, open datasets, data visualizations, and visual analytic dashboards (Leung et al., 2020; Ulahannan et al., 2020). However, critical interpretation of and reflections on data can also be weaponized to support and spread anti-scientific claims online. For example, anti-maskers on Twitter and Facebook leveraged data in similar ways as scientists to promote their arguments by identifying bias in data, participating in visualization creation and interpretation, and critically evaluating data representation and sources (Lee et al., 2021). Indeed, conspiracists have long used seemingly critical and scientific approaches to make their arguments convincing. Even before the COVID-19 pandemic, troll accounts and bots on Twitter strategically promoted vaccine hesitancy by broadcasting both pro- and anti-vaccine content, creating the illusion of a debate rather than a public health agenda (Broniatowski et al., 2018). Additionally, anti-vaccine campaigns

on social media have produced so-called "educational" materials that distort reputable epidemiological data and present it as scientific evidence (Hoffman et al., 2019).

What textual cues can support meaningful distinctions between such anti-scientific discourses about data and those based in more reliable, trustworthy evidence? Previous research has extensively applied machine learning approaches to detect anti-vaccine discourses on social media, including COVID-19 vaccines. One recent study achieved high performance using BERT to distinguish misinformation from general vaccine-related discussions (Hayawi et al., 2022). Beyond the context of vaccines, other studies have reported positive results using machine learning to distinguish misinformation about COVID-19 (Cui and Lee, 2020; Zhou et al., 2020; Abdul-Mageed et al., 2021; Abdelminaam et al., 2021). However, despite these promising outcomes, machine learning-based methods often fail to identify human-interpretable textual features. This limitation poses difficulties for researchers and platform developers to fully comprehend the underlying mechanisms and derive actionable insights, and consequently, there has been few effective educational tools on platforms that help people consume and make valid data-driven arguments while staying away from conspiracist discourses.

Different from these previous studies, we propose a theory-driven approach to distinguish anti-vaccine data discourses on social media. Drawing on prior literature about conspiracist communication from other settings and domains, we derive hypotheses for interpretable textual features that might characterize of conspiracist arguments. We then operationalize, measure, and compare these features in a sample of pro- and anti-COVID-19 vaccine data discourses taken from Twitter during the first six months in which COVID-19 vaccines were available (October 2020–April 2021). There are three features of conspiracist communication that we compare in pro- and anti- COVID-19 vaccine data discourses: mentions of authority figures (especially scientific and political leaders), expressions of certainty, and causal claims. Prior work suggests that all of these attributes occur more (often) in conspiracist rhetoric (Hoffman et al., 2019; Harambam and Aupers, 2021; Chen et al., 2023; Hullman et al., 2019; Pipes, 1997; Barkun, 2013), and we hypothesized analogous findings here (i.e., that the conspiracist discourse features would appear more in anti-vaccine data discourses). We evaluate our hypotheses using a series of regression models, and find mixed support: anti-vaccine COVID-19 data discourses in social media possessed more expressions of certainty and causal claims, but fewer mentions of authority figures.

While some of the salient prior work studied conspiracist communication in social media related to the COVID-19 pandemic, we are not aware of any that have constructed computational measures of these features, applied them in the context of COVID-19 vaccine data discourses in social media, or evaluated whether they distinguish pro- and anti-vaccine data discourses in this kind of setting. Our work contributes along all of these dimensions, advancing an approach as well as a set of interpretable distinctions that differentiate pro- and anti-vaccine data discourses.

5.2 Methods

5.2.1 Data collection

In order to evaluate whether features of conspiracist discourse differentiate COVID-19 pro- and anti-vaccine data discourse in social media, we constructed an original sample of tweets that contain discussions of COVID-19 data and express pro- or anti- vaccine sentiment between October 18^{th}, 2020 and April 1^{st}, 2021. This period captures the first 165 days in which COVID-19 vaccines were available to the general public. We used the v2 full-archive search endpoint in Twitter Academic API to collect tweet data.[2]

To identify tweets to include in the study, we created an initial list of pro-vaccine hashtags based on the hashtags used in studies about pro- and anti- vaccine discussions on Twitter during and prior to the COVID-19 pandemic (Milani et al., 2020; Hoffman et al., 2021). We also expanded on the initial list by searching among tweets that contained the hashtags in the initial list and looked for any new pro-vaccine hashtags that were not in our list. The final set contained 18 pro-vaccine hashtags such as "#provax", "#vaccineswork", and "#whyIvax". We followed similar steps to build the list of hashtags to identify anti-vaccine tweets. We referred to previous literature that studied anti-vaccine and vaccine hesitancy discourses on Twitter (Muric et al., 2021; Gunaratne et al., 2019; Blankenship et al., 2018; Milani et al., 2020; Hoffman et al., 2021) and used snowball sampling to expand the list. The final set contained 65 anti-vaccine hashtags such as "#antivax", "#vaccinesdontwork", and "#naturalimmunity". The complete lists of hashtags appears in Table B.2 in Appendix B.

To sample tweets that contained discussions about data, we referred to a previous study that focused on

[2] `https://developer.twitter.com/en/docs/twitter-api/tweets/search/introduction` (permalink: `https://perma.cc/U4WB-7FBB`)

data visualizations among anti-maskers on Twitter (Lee et al., 2021) and adopted the list of keywords that the authors used to sample tweets about data visualizations, including "chart(s)", "dashboard(s)", "map(s)", "plot(s)", "viz", "vis", and "visualization(s)". Because we were interested in broader discussions about data, we added the keywords of "data", "stats", and "statistics". To ensure high precision, we validated our sample by randomly selecting 10 tweets for each keyword and manually coding whether the tweet contained any discussion of data. The results of our validation analysis are shown in Table B.1 in Appendix B. We dropped the keywords "plot(s)", "viz", and"vis" due to low precision and used the rest of the keywords as the final list.

We constructed Tweet search queries using the pro- and anti- vaccine hashtags as well as the keywords about data. We dropped tweets that contained both pro- and anti-vaccine hashtags. Our approach emphasizes precision over recall: the resulting sample that allows us to evaluate features that may differentiate pro- and anti-vaccine tweets engaging in data discourse rather than all available tweets discussing vaccine data. Our final dataset contained $1,658$ tweets with pro-vaccine hashtags and $4,031$ tweets with anti-vaccine hashtags.

5.2.2 Hypotheses

The first hypothesis centers on the *mention of authority figures*. Previous research on conspiracists suggests that they tend to mention authorities, particularly individuals in science and politics, to support their claims (Harambam and Aupers, 2021). In anti-vaccine campaigns, a common strategy is to call out government and scientific entities and claim that they provide evidence for the negative impact of vaccines (Hoffman et al., 2019). This phenomenon is also observed in conspiracy discourses related to COVID-19, where anti-maskers often tried to support their points by referring to specific public figures in the scientific community (Lee et al., 2021). Anti-intellectualism is another reason that authority were often mentioned in conspiracy discourses. For example, conspiracy discourses about the COVID-19 pandemic often bring up researchers and try to delegitimize their motivation and knowledge (Chen et al., 2023). Based on these observations, we hypothesized that: *H1:* Anti-vaccine data discourses are more likely to mention authority figures than pro-vaccine data discourses.

Prior work has also found that conspiracist discourses tend to express greater *certainty*, ignoring the uncertain nature of science. In general, failing to communicate uncertainty can make scientific arguments

seem less trustworthy, which might be desirable for those expressing anti-vaccine perspectives (Hullman et al., 2019). Therefore, we hypothesized that: *H2:* Anti-vaccine data discourses tend to appear as more certain than pro-vaccine data discourses.

Finally, conspiracist discourses tend to make *causal claims*, even when such relationships may not exist or are only correlational (Pipes, 1997). This is because that conspiracy theorists tend to allegedly reduce "highly complex phenomena to simple causes" and spend more energy promoting their theories rather than explaining it (Barkun, 2013). Based on these observations, we hypothesized that: *H3:* Anti-vaccine data discourses are more likely to contain causal claims than pro-vaccine data discourses.

5.2.3 Measures

Tweets are the units of our analysis. We labeled each tweet as "pro-vaccine" or "anti-vaccine" based on the pro- or anti-vaccine hashtags it contained. We then constructed a binary variable *Is.pro-vaccine* as the independent in our analysis.

We constructed the following dependent variables and control variables to test our three hypotheses.

Dependent Variables

H1: Mention of authority figures. To investigate H1 about appeals to authority, we identified tweets that named entities whose public role had some relevance to COVID-19 vaccines: scientific researchers, political officials, and physicians. We thus derived three sub-hypotheses from H1:

1. H1.1: Anti-vaccine data discourses are more likely to mention researchers than pro-vaccine data discourses.

2. H1.2: Anti-vaccine data discourses are more likely to mention politicians than pro-vaccine data discourses.

3. H1.3: Anti-vaccine data discourses are more likely to mention physicians than pro-vaccine data discourses.

To construct the dependent variables for H1, we first applied the Named Entity Detection method in the SpaCy library[3] to identify any named entities in the tweets in our sample. Name entities can include

the names of individual persons, organizations, or Twitter account handles. In cases where the detected entity was a Twitter account handle, we queried Twitter to get the name associated with the account. Using the resulting list of individual and organizational names mentioned in tweets, we collected information about each named entity by querying the Wikidata database[4] using its API. In particular, we labeled each entity as "researcher", "physician", "politician" based on the data from the "occupation" field in Wikidata entries. We then constructed dichotomous variables *Mentioned.researcher?*, *Mentioned.politician?*, and *Mentioned.physician?* for H1.1., H1.2, and H1.3 respectively. To test H1, we also constructed another dichotomous variable, "Mentioned.authority", that captures whether a tweet mentions any of the three types of public figures.

H2: certainty. For H2, we use the Linguistic Inquiry and Word Count (LIWC) dictionary, the largest such resource in English, to identify features in tweet text that indicated certainty[5]. LIWC has been widely used to analyze sentiment in social text data (Tausczik and Pennebaker, 2010). The LIWC dictionary categorizes common English words into many sentiment categories, including "certainty", which we use as a proxy here. We constructed a dichotomous variable *Is.certain?* that captures whether a tweet contained any words or phrases in the "certainty" category of LIWC.

H3: Causal claims. Lastly, to investigate H3 about casual claims, we used the same dictionary-based strategy from H2, which has also been employed in previous literature investigating causal claims in Twitter discussions. The LIWC dictionary contains another category for "causal" sentiments and wee constructed a dischotmous variable *Is.causal?* to capture whether a tweet contained any words or phrases from this LIWC category.

Control Variables

We also included the following control variables in our analysis as they capture features of tweets or Twitter accounts that could help to explain variation in the dependent variables.

[3] `https://pypi.org/project/spacy/` (permalink: `https://perma.cc/M3V4-5UP4`)
[4] `https://www.wikidata.org/wiki/Wikidata:Main_Page` (permalink: `https://perma.cc/KTA2-S32U`)
[5] `https://www.liwc.app/` (permalink: `https://perma.cc/5YCQ-SPDY`)

Tweet Length. Longer tweets contain more words than a shorter tweet, and therefore may be more likely to contain named entities or more words related to certainty and causal claims. The variable *Tweet.length* is a count of the number of words in each tweet.

Account history. The experience of an user in a community can affect their discourses. For example, a new user may be more likely to use certain rhetoric strategies to attract attention, and an experienced user may tend to use some other rhetoric strategies to maintain audiences. To account for these factors, we constructed a control variable *Account.tweet.count* as a proxy to measure the history of an account and in other words, the experience of the user. We counted the total number of published tweets an account in our dataset had posted at the time of our data collection (November 2022). Since Twitter does not provide historical data about individual user accounts, we were not able to collect the number of published tweets an account had when the account posted the tweet that we collected (i.e., between October 2020 and April 2021). However, since previous literature indicates that the total number of tweets an account posts tend to increase in a linear manner over time , we believe that the distributions of number of total tweets in our dataset in November stay the same as the historical distribution.

Size of audience. Additionally, the way an author makes arguments or discourses in a tweet can be influenced by their imaginary audience. To account for the effect of the imaginary audience of a tweet, we constructed a control variable *Account.follower.count* by counting the number of followers an account in our dataset had at the time of our data collection (November 2022). Similar to our construction of the last control variable, we were not able to collect the number of followers an account had when the account posted the tweet that we collected (i.e., between October 2020 and April 2021). Similarly, as previous literature suggest, the number of followers of a Twitter account tends to grow linearly over time. We thus believe that the distributions of number of followers of the accounts in our dataset in November stay the same as the historical distribution.

There were 748 accounts deleted or suspended at the time of our data collection, so we were not able to obtain data about their follower number and total number of tweets that the account ever posted. We used the median numbers for these missing numbers in the following analysis.

5.2.4 Analysis

H1

To test H1a using the binary outcome variable *Mentioned.authority?* for H1, we fit a logistic regression model on the dataset of tweets using the GLM function in R.[6] Because the distribution of the count control variables *Account.tweet.count*, *Account.follower.count*, and *Tweet.length* are right-skewed, we use a started log transformation in all the models involving these controls (i.e., $\log(x+1)$). Our formal model for H1 (referred as M1) is as follows:

$$\log\left(\frac{\hat{p}(Mentioned.authority?)}{1-\hat{p}(Mentioned.researcher?)}\right) = \beta_0 + \beta_1 Is.pro.vaccine?$$
$$+ \beta_2 \log(Account.tweet.count+1) + \beta_3 \log(Account.follower.count+1) + \beta_4 \log(Tweet.length+1)$$

To test H1.1, we fitted a model (M1.1) that is identical to that of H1, except using the variable *Mentioned.researcher?* to construct the outcome of the logistic regression model:

$$\log\left(\frac{\hat{p}(Mentioned.researcher?)}{1-\hat{p}(Mentioned.researcher?)}\right)$$

Similarly, to test H1.2, we fitted a similar model (M1.2) using the variable *Mentioned_politician?* to construct the outcome of the logistic regression model:

$$\log\left(\frac{\hat{p}(Mentioned.politician?)}{1-\hat{p}(Mentioned.politician?)}\right)$$

Finally, to test H1.3, we fitted a model (M1.3) that is identical to that of H1, except using the variable *Mentioned_physician?* to construct the outcome of the logistic regression model:

$$\log\left(\frac{\hat{p}(Mentioned.physician?)}{1-\hat{p}(Mentioned.physician?)}\right)$$

H2

To test H2, we fitted a model (M2) that is identical to those of H1, except using the variable *Is_certain?* to construct the outcome of the logistic regression model:

$$\log\left(\frac{\hat{p}(Is.certain?)}{1-\hat{p}(Is.certain?)}\right)$$

H3

[6] `https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/glm` (permalink: `https://perma.cc/EP9Q-MS8D`)

To test H3, we fitted a model (M3) that is identical to those of H1, except using the variable *Is_causal?* to construct the outcome of the logistic regression model:

$$\log\left(\frac{\hat{p}(Is.causal?)}{1-\hat{p}(Is.causal?)}\right)$$

5.3 Results

The dataset we analyze consists of 5,689 tweets containing discussion of data about the COVID-19 vaccine. This includes 1,658 pro-vaccine and 4,031 anti-vaccine data discourse tweets.

First, we summarize the features of the pro- and anti-vaccine data discourse tweets used in our analysis in Table 5.1. Figure 5.1 presents bivariate barplots of our outcomes within strata representing pro- and anti-vaccine data discourse tweets. Then, we present the results of our regression analyses in Table 5.2 as well as model-predicted probabilities in Figure 5.2. Details about the construction of all variables, hypothesis tests, analyses, and model specifications appear in the Methods section. Across all measures and analyses, we observe substantial differences between the pro- and anti-vaccine data discourse tweets.

Of the pro-vaccine data discourse tweets, 3.50% mention at least one authority figure, including 0.84% that mentioned researchers, 2.41% that mentioned politicians, and 0.90% that mentioned physicians. Within this same pro-vaccine set, 21.05% expressed certainty and 33.17% contain casual claims. The median length of pro-vaccine data tweets is 253 characters, the median number of followers for accounts posting such tweets is 1,364, and the median total number of tweets ever posted is 10,490.

Of the anti-vaccine data discourse tweets, a smaller proportion (0.57%) mention at least one authority figure, including 0.10% that mentioned researchers, 0.42% that mentioned politicians, and 0.17% that mentioned physicians. Higher proportions expressed certainty (31.61%) and contained casual claims (39.35%). The median length of anti-vaccine data discourse tweets is 275 characters, the median number of followers for accounts posting such tweets is 920, and the median total number of tweets ever posted is 12,250.

5.3.1 Mention of authority figures

Pro-vaccine data discourses on Twitter were more likely to mention authority figures than anti-vaccine data discourses. This is contrary to the direction of difference we hypothesized (see H1 in Methods). This was also the case for the specific subgroups of authority figures salient to the COVID-19 pandemic (researchers,

	Data type	Pro-vaccine data tweets			Anti-vaccine data tweets		
		Mean	St. Dev.	Median	Mean	St. Dev.	Median
Mentioned.authority?	binary	0.03	-	0.00	0.01	-	0.00
Mentioned.researcher?	binary	0.01	-	0.00	0.00	-	0.00
Mentioned.politician?	binary	0.02	-	0.00	0.00	-	0.00
Mentioned.physician?	binary	0.01	-	0.00	0.00	-	0.00
Is.certain?	binary	0.21	-	0.00	0.32	-	0.00
Is.causal?	binary	0.33	-	0.00	0.39	-	0.00
Tweet.length (# of characters)	count	235.76	70.07	253.00	258.32	80.76	275.00
Account.follower.count	count	37835.52	505130.66	1364.00	57479.07	1656239.95	920.00
Account.tweet.count	count	29072.05	84659.67	10490.00	40555.02	125952.44	12250.00

Table 5.1: Summary of pro and anti-vaccine data discourse tweets in our dataset

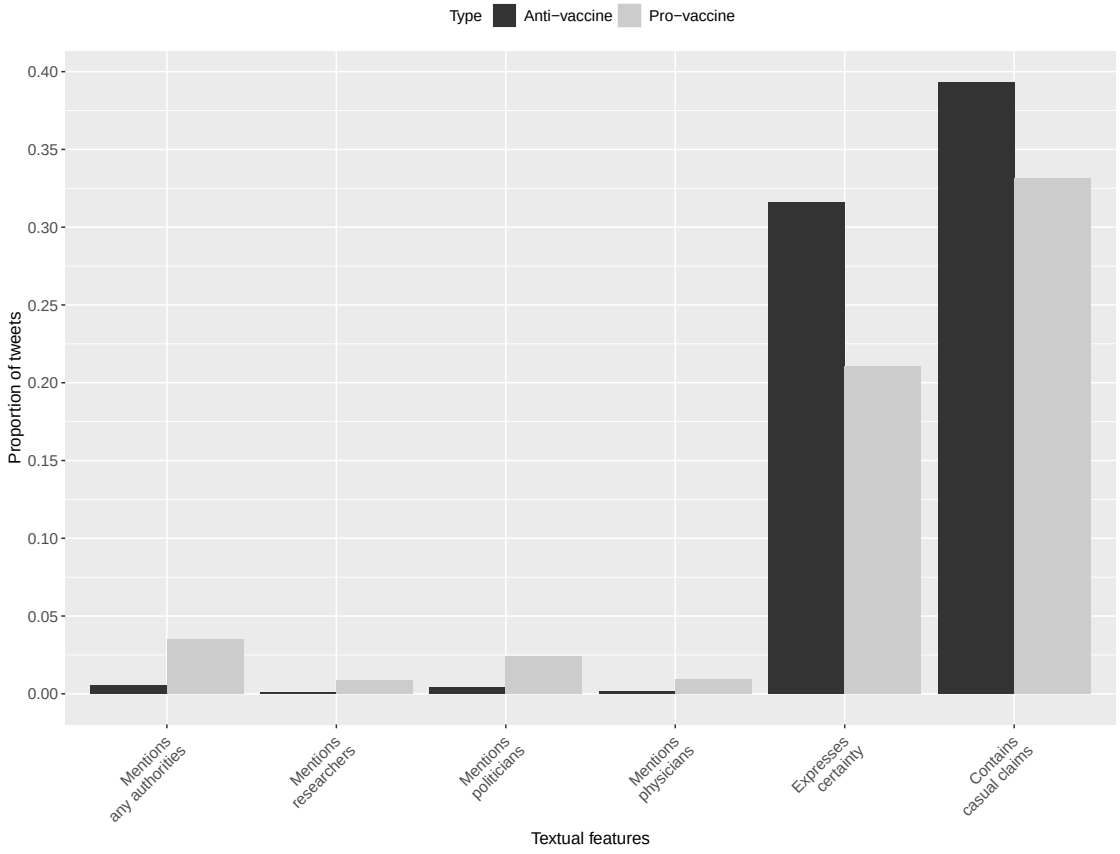


Figure 5.1: Bivariate barplot of the distributions of outcome variables across pro- and anti-vaccine data discourse tweets.

politicians, and physicians) as pro-vaccine data discourses on Twitter were more likely to mention all three. Model M1 in Table 5.2 estimates that a pro-vaccine data tweet was about 8 times more likely to mention an authority figure than an otherwise similar anti-vaccine data tweet ($\beta = 2.08$, SE $= 0.26$, $p < 0.000$). Models M1.1–M1.3 show that a pro-vaccine data tweet was about 11 times more likely ($\beta = 2.36$, SE $=$

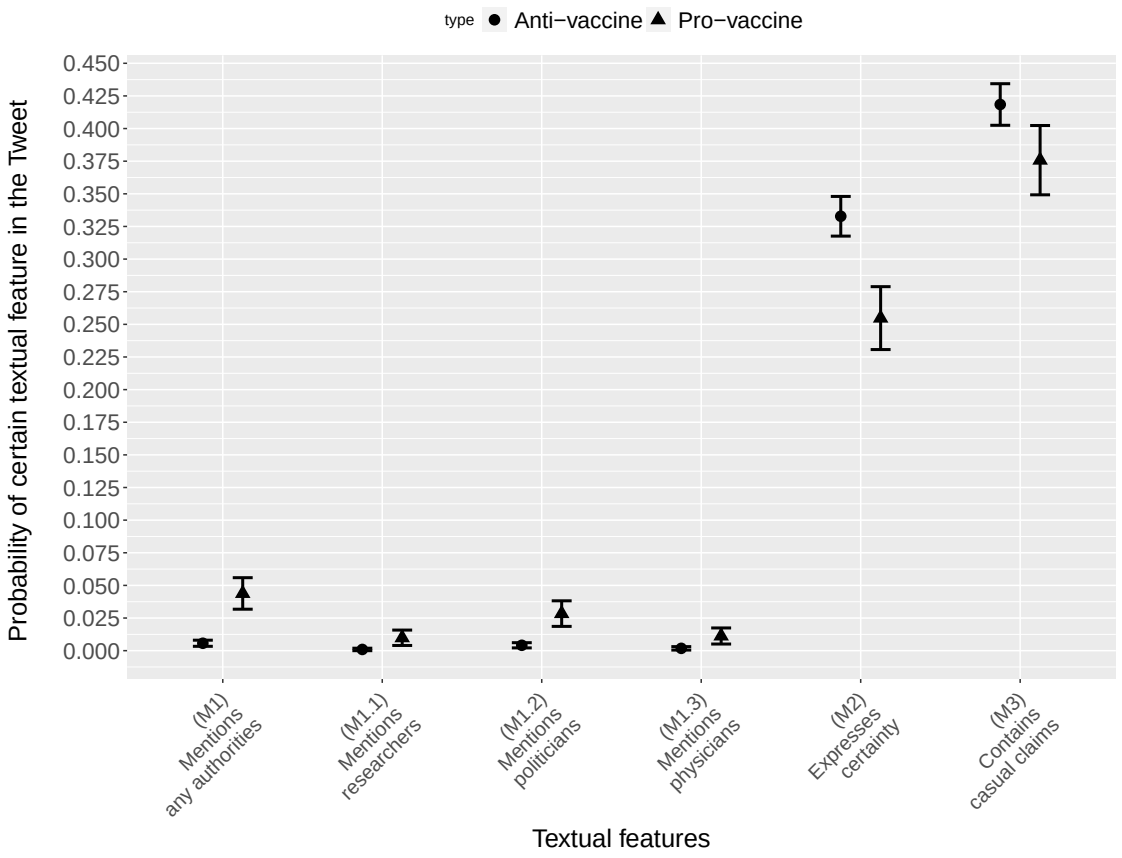


Figure 5.2: Model predicted probabilities corresponding to our hypotheses. Each pair of data points shows the predicted probability for two tweets who have sub-sample median values for each of our control variables; these tweets vary only in terms of whether the tweet is pro-vaccine or anti-vaccine. Error bars reflect the marginal effects of our key independent variable ($1.96 \times \text{SE}$).

0.60, $p < 0.000$), 7 times more likely ($\beta = 1.95$, $\text{SE} = 0.31$, $p < 0.000$), and 6 times more likely ($\beta = 1.86$, $\text{SE} = 0.48$, $p < 0.000$) to mention researchers, politicians, and physicians respectively than an otherwise similar anti-vaccine data tweet.

In terms of model-predicted prototypical tweets, Figure 5.2 illustrates that 4.38% of the prototypical data tweets that are pro-vaccine would mention authority figures, whereas 0.57% of otherwise similar anti-vaccine data tweets would. M1.1 predicts that 0.99% of the prototypical data tweets that are pro-vaccine would refer to researchers, whereas 0.09% of otherwise similar anti-vaccine data tweets would. Similarly, M1.2 predicts that 2.84% of the prototypical data tweets that are pro-vaccine would refer to politicians, whereas 0.41% of otherwise similar anti-vaccine data tweets would. Furthermore, M1.3 predicts that 1.12% of the prototypical data tweets that are pro-vaccine would refer to physicians, whereas 0.18% of otherwise

	M1	M1.1	M1.2	M1.3	M2	M3
(Intercept)	-11.73^{***}	-19.77^{***}	-8.68^{***}	-13.33^{**}	-8.72^{***}	-8.42^{***}
	(2.28)	(4.43)	(2.61)	(4.24)	(0.65)	(0.60)
Is.pro-vaccine?	2.08^{***}	2.36^{***}	1.95^{***}	1.86^{***}	-0.38^{***}	-0.18^{**}
	(0.26)	(0.60)	(0.31)	(0.48)	(0.07)	(0.07)
log1p(Account.tweet.count)	0.07	-0.04	0.04	0.10	0.05^{*}	-0.03
	(0.08)	(0.17)	(0.10)	(0.16)	(0.02)	(0.02)
log1p(Account.follower.count)	-0.13	0.01	-0.15	-0.07	-0.06^{**}	0.02
	(0.07)	(0.14)	(0.08)	(0.13)	(0.02)	(0.02)
log1p(Tweet.length)	1.22^{**}	2.34^{**}	0.68	1.18	1.42^{***}	1.47^{***}
	(0.40)	(0.77)	(0.45)	(0.73)	(0.11)	(0.10)
AIC	782.65	226.52	600.41	280.89	6549.56	7276.41
BIC	815.88	259.75	633.65	314.12	6582.79	7309.65
Log Likelihood	-386.33	-108.26	-295.21	-135.45	-3269.78	-3633.21
Deviance	772.65	216.52	590.41	270.89	6539.56	7266.41
Num. obs.	5689	5689	5689	5689	5689	5689

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 5.2: Logistic regression models that, respectively, predict the probability that a tweet mentions any general authority figures (M1), researchers (M1.1), politicians (M1.2), physicians (M1.3), expresses certainty (M2), and includes causal claims (M3). The models are fit to the tweet-level dataset that includes 5689 tweets.

similar anti-vaccine data tweets would.

5.3.2 Certainty

We found support for H2. Our analysis shows that, when compared to anti-vaccine tweets, pro-vaccine tweets were significantly less likely to involve expressions that are certain. We present the results of our regression analysis in Table 5.2. We found that the odds that a pro-vaccine data tweet would include words that indicate certainty are only 0.69 of an otherwise similar anti-vaccine data tweet ($\beta = -0.38, \text{SE} = 0.07$, $p < 0.000$). We further explain the results of our regression analysis in terms of prototypical data tweets. As seen in Figure 5.2, our model predicts that only 25.48% of the prototypical data tweets that are pro-vaccine would include words that indicate certainty, whereas 33.28% of otherwise similar anti-vaccine data tweets would.

5.3.3 Causal claims

We were also able to find support for H3 in our analysis. We found that when compared to anti-vaccine tweets, pro-vaccine tweets were significantly less likely to include causal claims. We present the results of our regression analysis in Table 5.2. We found that the odds that a pro-vaccine data tweet would include

words that indicate casual claims are only 0.84 of an otherwise similar anti-vaccine data tweet ($\beta = -0.18$, $\text{SE} = 0.07$, $p < 0.007$). We also illustrate the results of our regression analysis in terms of prototypical data tweets. As seen in Figure 5.2, our model predicts that 37.58% of the prototypical data tweets that are pro-vaccine would include words that indicate certainty, whereas 41.84% of otherwise similar anti-vaccine data tweets would.

5.4 Discussion and conclusion

While the public's growing critical engagement with vaccine data in social media could be vulnerable to conspiracy claims, our study points to a promising pathway to mitigate the impact of conspiracy data discourses by identifying and distinguishing online conspiracy arguments that misuse data. Focusing on the specific context of the COVID-19 global pandemic, our research investigated online discussions regarding vaccine data on Twitter. we propose a theory-driven approach to distinguish anti-vaccine social media discussions of data by identifying textual characteristic of conspiracy discourses. While previous research has extensively explored machine learning approaches to detect conspiracist discourses about vaccines on social media, our theory-driven hypothesis testing approaches provide insights that may explain the success of prior machine learning approaches to distinguishing information integrity in similar natural language settings. Adding to the considerable effort has been made to combat vaccine misinformation on online social platforms prior to the COVID-19 pandemic, including fact-checking mechanisms and platform regulation methods (Burki, 2019), our study outlines a new approach for online social platforms to safeguard the public against conspiracy discourses.

Our results indicate that anti-vaccine data discourses on Twitter are more likely to exhibit certainty and make causal claims, compared to pro-vaccine tweets engaging with data. This finding aligns with prior literature on conspiracy theories and online misinformation in other domains. Our study suggests that the presence of certain textual features, such as certain sentiment and the claims of causal relationships, can be crucial factors in identifying conspiracy data discourses. Although our current approach to capturing these textual features is relatively simple (i.e., presence of single words or phrases indicating certainty or causal claims based on a specific sentiment dictionary) and the proxies we used to construct the variables in our models may be noisy, future research could explore more complex, computationally intensive approaches.

Our findings also have implications for online social platforms. On the one hand, platforms can use these textual features to initially filter out potential conspiracy discourses among discussions about data, and apply closer moderation afterwards. In addition, platforms could implement guidelines for public engagement with data, encouraging users to be less certain and exercise caution when drawing causal conclusions in their interpretations and arguments related to data.

Surprisingly, our findings regarding the mention of authority figures contradict our initial hypothesis and prior literature on conspiracy theories. We found that pro-vaccine tweets about data are more likely to mention authority figures from a range of domains, including researchers, politicians, and physicians, compared to anti-vaccine tweets involving data. Although the specific mechanism behind this phenomenon remains unclear, we speculate that it may be related to the abundance of up-to-date COVID-19 information regarding vaccines on social media, that everyone of the general public can refer to what authorities have said and done. Future research could investigate the mechanism behind this phenomenon in depth, exploring topics such as the specific aspects of authority figures that people discuss across different domains and how that contributes to their arguments about data. Interestingly, our findings suggest that people are adopting strategies previously associated with conspiracy theorists to potentially counter conspiracy data discourses. While we did not study the mechanism of this observation, it might indicate that individuals are intentionally employing the opposing side's rhetoric tactics to persuade and convince each other. Platforms could leverage this trend by encouraging users to refer to and critically reflect on authoritative sources when making and consuming data-driven arguments. For example, they could implement features that highlight verified information and provide easy access to reputable sources alongside user-generated content. Additionally, platforms might introduce tools that guide users in evaluating and prompt them to engage in discussions about the credibility of different sources. Future researchers could also dive deeper into the reasons and motivations behind individuals adopting these rhetorical strategies.

Overall, the findings of this study offer insights to the challenges and opportunities for promoting responsible engagement with data in online social spaces. Future research could further explore and validate the set of textual features that we identified in contexts beyond online discourse related to COVID-19 vaccines. We encourage researchers and online platform developers to address the issues of conspiracy claims in data discourses and to cultivate a better environment for the public to critically engage with vaccine data.

5.5 Summary

While online communities and social media platforms offer an open landscape for individuals to critically engage with data relevant to them and their communities, it is essential not to overlook the potential pitfalls of such engagements becoming muddled by conspiracy discourses. My study demonstrates that it is possible to differentiate between the practices of critical data literacy and conspiracist narratives, and that we should seek a new layer of critical data literacy specific to informal online settings.

Previous research on critical data literacy emphasized on fostering critical reflection on the context and ethics in the collection, processing, analysis, and decisions made out of data. This study, adding on to this body of work, recognizes that these critical engagement are situated in a social information landscape, where discourses about data made by one member are actively shared, interpreted, and shaped by other members. Thus, when facilitating critical data literacy in informal settings, whether through designing platform features or implementing regulation policy, we should guide individuals to not only focus on the data but also on how arguments surrounding it are constructed. Beyond just the data, they should be scaffolded to critically reflect on the narrative or the meta delivery of the arguments. The textual and rhetorical features highlighted in this study, such as causal claims, certainty, and references to authority, serves as examples of the facets of critical data discourse that should be scrutinized. Undoubtedly, there are many more such facets to consider.

Chapter 6

Study D: Dataland

Prologue

This study was conducted in 2022. While Study A examines how understandings of specific programming concepts about data are shaped within an informal social learning space, it represents only a glance of the myriad ways novices can computationally engage with data. There are broader dimensions to this, as outlined by D'Ignazio and Bhargava (2015), including reading, working with, analyzing, and arguing with data. Open questions include what programmatic practices novices tend to display while working with a dataset, what understanding they may form in this process, and how these computational engagements can shape their perspectives on data more broadly. Therefore, I pursued this study to construct a framework that describes the concepts, practices, and perspectives related to programming with data for novices in informal contexts.

As of 2022, few prevalent platforms were specifically designed for novices to program with data in informal settings. Recognizing this gap, I have engaged in designing and developing a research system to support novices in actively writing computer programs to make sense of data. I worked as a core member of a team developing *Dataland*[1]—a visual block-based programming system consisting of code editors and data tables embedded in a web-based story interface, with the goal of scaffolding novices to process, analyze, and visualize data. I conducted research workshops with 28 young participants to study their practices of working with data scaffolded by *Dataland* and developed a framework of computational data literacy.

[1] `https://learning-with-data.github.io/` (permalink: `https://perma.cc/TBU2-VWH7`)

This framework describes the concepts, practices, and perspectives that are key in novices using computer programming to perform data processing, analysis, and visualization, as well as the design implications on how to support novices in developing computational data literacy.

The subsequent sections of §6.1 to §6.7 in this chapter feature the paper I published at the ACM Interaction Design for Children conference in 2023 (IDC'23) (Cheng et al., 2023a). The paper was a collaborative effort between myself, Aayushi Dangol, Frances Marie Tabio Ello, Lingyu Wang, and Sayamindu Dasgupta. I was the lead of the work and participated in the design of the system, including the creation of one of the storyline interfaces, the design and execution of workshops, the collection and analysis of data collected in the workshops, and the writing of the manuscript. The content of the paper remains in its original form without any modifications. §6.8 provides a short summary of the takeaways of this study in relation to the thesis of this dissertation.

6.1 Introduction

Being able to make sense of our world through data has become an increasingly important ability. Significant interest has emerged in approaches that could support the development of this ability in middle- and high-school-aged young people (Lee and Wilkerson, 2018; Rubin, 2020). Often referred to as *data literacy*, a key requirement of this skill is to become familiar with practices around data, such as reading data, working with data, analyzing data, and arguing with data (D'Ignazio and Bhargava, 2015). A subset of these practices is often realized through computational means, commonly by writing computer programs. For example, analyzing a large dataset typically involves writing a program to filter and reshape the data, which can then be used as input for another bespoke program that generates a visualization. Compared to pre-programmed tools (e.g., Microsoft Excel) for data processing, programming with data offers a wider array of possibilities (e.g., new forms of data visualization) (Dasgupta and Hill, 2017b; Hill et al., 2017). Although programming with data is a practice seen primarily among professionals (e.g., professional data analysts) (Kandel et al., 2012; Kross and Guo, 2019), recent discussions of democratizing data science call for everyone, including young people, to learn computational skills that support asking and answering questions with data (Hill et al., 2017).

This paper showcases our attempt to bring the power of programming with data to children. We focus on

the computational aspects of data literacy, which we refer to using the term *computational data literacy* (Yalcinkaya et al., 2022). We first introduce *Dataland*, a visual block-based programming system designed for young people that focuses on data analysis and visualizations situated in story contexts. We describe the system and then report on workshops where children used the programming system and played the role of an investigator to conduct a series of data analyses in order to help the story's protagonist locate a missing family member. We build on Brennan and Resnick (2012a)'s framework of Computational Thinking and present our findings as a framework for studying computational data literacy. We conclude with a discussion of our findings, including recommendations for future designers of systems to support the development of computational data literacy. Our contributions are as follows.

- A visual block-based programming system to analyze and visualize data in the context of narrative storylines.
- Empirical findings from workshops where young people used the system.
- An extension of an existing framework that presents computational data literacy in terms of three distinct dimensions—concepts, practices, and perspectives.
- Recommendations for future designers of systems that support the development of computational data literacy.

6.2 Related work

6.2.1 Computational data literacy: youth data literacy and computational thinking

In today's data-driven world, the ability to understand and use data has become crucial not only for professional data scientists but also for everyday people (D'Ignazio and Klein, 2020; Hill et al., 2017). Argued by many to be a new form of literacy, *data literacy* has been brought up as an essential skill that young people should acquire (Lee and Wilkerson, 2018). Existing scholarly work has explored and proposed different dimensions of data literacy. Many agree that data literacy for young people includes the ability to read data (e.g., understanding what a dataset represents), work with data (e.g., cleaning a dataset), analyze data (e.g., filtering data), and argue with data (e.g., visualizing data to support a claim) (D'Ignazio and Bhargava,

2015; Tygel and Kirsh, 2016). Data science education research also emphasizes the ability to understand the context and human factors that affect data collection and analysis, as well as to aggregate, visualize, and make inferences with data (Rubin, 2020). Additionally, increasing attention has been paid to the critical aspects of data literacy, (e.g., the ability to decipher how data is created) (Dourish and Gómez Cruz, 2018; Feinberg, 2017; Wilkerson et al., 2021), as well as the skills to access and interpret data through the lens of community and social impact (Elisa Raffaghelli, 2020; Hautea et al., 2017a; Matuk et al., 2020).

Certain aspects of data literacy are associated with the learner's ability to program with data. For example, one can write programs to efficiently and reliably sort, filter, clean, join, and make meaningful visualizations of large and complex datasets (Weintrop et al., 2016). In this paper, we focus on programming with data, and in order to differentiate this from broader data literacy that may not involve programming, drawing from Yalcinkaya et al. (2022), we call the ability to work with data through programming *computational data literacy.* Apart from data literacy, computational data literacy also connects to the broader notion of Computational Thinking (CT), the ability to solve problems "by drawing on the concepts fundamental to computer science. (Wing, 2006, p. 33)" CT has been studied and theorized extensively. For example, Brennan and Resnick (2012a) offer a well-known framework for CT that decomposes it into a series of *concepts* such as sequences, loops, and conditionals; *practices* such as testing, debugging, and abstracting; and *perspectives* of what CT enables, such as expressing oneself, connecting with others, and questioning technology. In more recent work that is closer to our topic, Basu et al. (2020) identifies a set of "focal knowledge, skills, and abilities" for assessing the concept of "Data and Analysis" as defined by another CT framework (Parker and DeLyser, 2017), and Berikan and Özdemir (2020) highlighted that "problem solving with datasets" as a key implementation of CT. Our work builds on these prior works and contributes a complementary series of concepts, practices, and perspectives that are unique to computational data literacy.

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

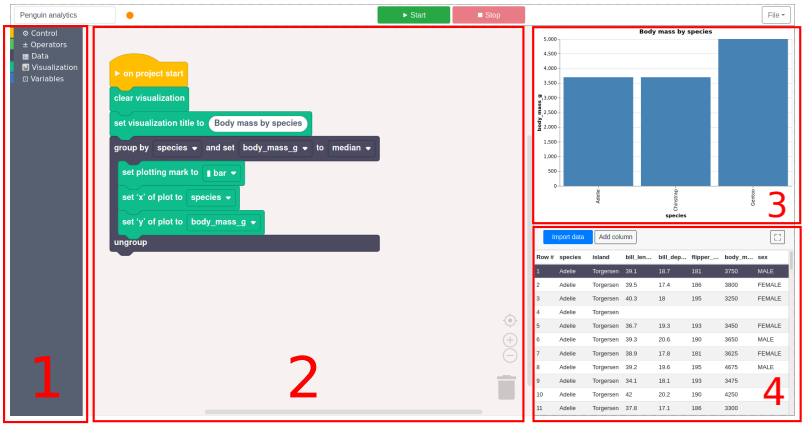
In recent years, a number of systems and scaffolds that engage young people in analyzing and visualizing data have emerged. For example, the Quantified Self movement—a global trend where sensors on mobile and wearable technologies are used to collect data on one's own everyday activities—has been seen as a rich environment for learning with and about data (Lee, 2013). The Common Online Data Analysis Platform

(CODAP) (Finzer and Damelin, 2015) system allows students of grades 6–12 to access and explore data from a wide variety of sources. Dasgupta and Hill (2017b) designed Scratch Community Blocks, a system built on top of the Scratch programming environment and the online community (Resnick et al., 2009c) that allowed members of the Scratch community to programmatically access, analyze, and visualize their own activities in the community. In addition to inventing novel systems, researchers have also been developing toolkits, curricula, and community activities that support youth data literacy. Examples include real-world datasets prepared for a variety of educational contexts (Bart et al., 2017), as well as classroom and outreach activities developed to engage students and community members in recognizing bias and complexity in data and producing data artifacts for social impact (Bhargava et al., 2016; D'Ignazio, 2017; Deahl, 2014; D'Ignazio and Bhargava, 2015).

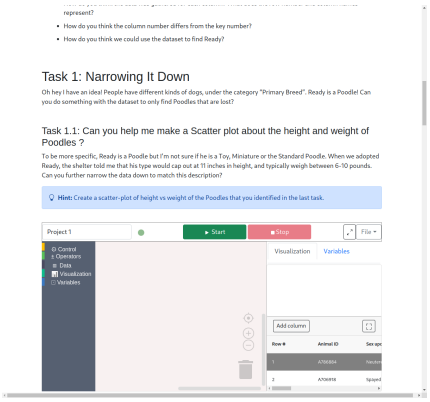
Despite the pluralism in data literacy tools, few systems specifically support computational data literacy in youth or directly support young people with limited programming experience to effectively program with data. While Scratch Community Blocks (Dasgupta and Hill, 2017b) allowed children to program with data using visual blocks, the data was limited to those of their activities in Scratch and the programming grammar and conventions were constrained to that of Scratch. Building on prior work and to explore the design space with fewer constraints, we present our design that supports youth computational data literacy. We draw design guidance from principles of Constructionism, a framework that has been widely used to design systems that support the learning of mathematical knowledge and programming skills (Papert, 1980a; Kafai, 2017). Following the Constructionist principles of "low floors" for novice entry, "high ceilings" for enabling advanced possibilities, and "wide walls" for exploration (Resnick, 2016), we designed a visual block-based programming system called *Dataland* that offers several built-in data query and visualization programming primitives. Inspired by prior work that recognizes the importance of situating data analysis within a broader narrative context (Clegg et al., 2020; Lee, 2013; Rubin et al., 2006; Register and Ko, 2020), we implement data stories in *Dataland* that position users in a personally engaging narrative.

6.3 System design

Dataland is presented through a "storyline," (Figure 6.1b) a web-based interface that tells a data story (described in §6.3.2). Throughout the storyline are several instances of code editors (shown in Figure 6.1a),



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



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

Figure 6.1: The *Dataland* code editor and story interface.

where users can work on block-based programming projects. In this section, we describe the code editor and the storyline interface. Readers can access the code editor on our project website, `https://learning-with-data.github.io/`.

6.3.1 Language design: Vocabulary and grammar

The *Dataland* code editor contains four main components: 1) a block palette containing all the programming blocks; 2) a coding area where users can program by dragging and dropping blocks; 3) a panel that shows visualization created by code and the dataset in the data table; and 4) a view of a data table that can be pre-populated or imported from a CSV file and edited (via code or the "Add column" button) by a user. The visualization system of the editor is implemented through *Vega-Lite* (Satyanarayan et al., 2017) and Leaflet.js,[2] and the block-based editor is implemented through the Blockly framework (Fraser, 2015). In Figure 6.1a and the rest of this section, we use the Palmer Penguins dataset (Horst et al., 2020) for illustrative purposes.

In *Dataland*, the programming blocks are the primitives that form the vocabulary of the programming language. In a way similar to how text is constructed with the vocabulary of a natural language, children who use *Dataland* will compose these primitives to create programs. The design of the *Dataland* programming

[2] `https://leafletjs.com/` (permalink: `https://perma.cc/A3GD-2CP5`)

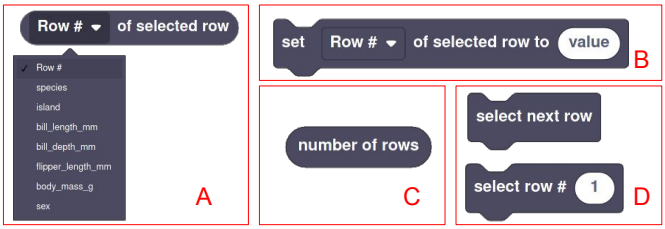


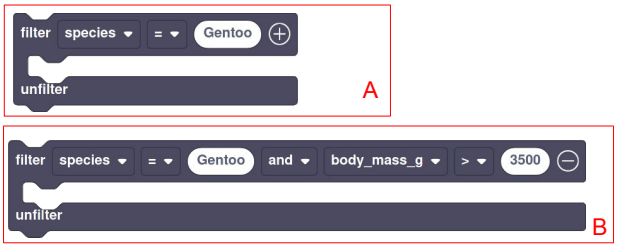
Figure 6.2: Blocks for accessing data: the Data access block with column drop-down (A); Block to set value in a row (B); Reporter block that returns total number of rows (C); Blocks for selecting data rows (D).

blocks follows principles in constructionist learning theories, where the choice of these building blocks "determines, to a large extent, what ideas users can explore with the kit—and what ideas remain hidden from view" (Resnick and Silverman, 2005, p. 119). We adopted much of the design of basic programming blocks (e.g., mathematical operations, conditions, loop) from the design of the Scratch programming blocks (Resnick et al., 2009c; Maloney et al., 2010) and added new blocks that are specific for data analysis. In this section, we focus on aspects of the language design that are unique to *Dataland*. Along with individual blocks (which represent the vocabulary of the language), we also provide an overview of the grammar—the rules that govern how the blocks can be composed. Drawing from Resnick and Silverman (2005), especially the principle of "inventing things that you would want to use yourself," the starting point of these design features are our experiences of working and educating with, as well as designing existing data programming systems and platforms.

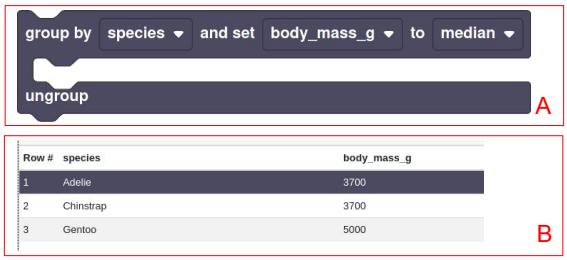
Navigating data table and accessing data

A fundamental requirement for any data programming system is to allow for accessing or reading data. Data is imported into *Dataland* through a user-interface gesture (a button click) or can be predefined in a storyline. After that, users can interact with imported data through programming. Following the design principles of "making execution visible" and "making data concrete" from Scratch (Maloney et al., 2010), during *Dataland*'s program runtime, a specific row is always in a selected state in the data table, and any changes in row selection and cell values are reflected in real time.

For programmatic access, we designed a set of unique data blocks (Figure 6.2). To change row selection, users will need the `select next row` or `select row # _` block (Figure 6.2 D). To access specific value from a row, the `_▼ of selected row` "reporter" block has a drop-down menu (indicated with



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



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

Figure 6.3: Filter and aggregation blocks.

the ▼ symbol) with a list of column names (Figure 6.2 A). This reported value can then act as an argument or input for another block (e.g., a mathematical operation block). Finally, a `number of rows` reporter block allows users to set up loops (Figure 6.2 C) to traverse the entire set of rows. In addition to access, users can also set specific values within a row with the `set _▼ of selected row to _` block (Figure 6.2 B). If a new column is needed (e.g., a derivative column), it can be added via a button click on the user interface.

Filtering

Filtering data in *Dataland* is made possible by the `filter` block that we designed. Filtering data is a common data analysis task that, at the conceptual level, builds on logical true-false predicates combined with Boolean operators, resulting in data transformations (e.g., a subset of the original table being returned). We designed the `filter` block as a c-shaped block, or "c-block", which is usually used to indicate contexts where temporary operations on the filtered data might take place. We further added labels "filter ..." and "unfilter" at the top and bottom of the block to suggest that the original dataset will be restored after the operations on the filtered data are complete (Figure 6.3a A). In addition, since it is common in data analysis tasks to specify variable-length conditions for filtering, combined with Boolean operators (e.g., $species == Gentoo\ AND\ body\_mass > 3500$), our design allows multiple predicates in the `filter` block (Figure 6.3a B). With this design, users can start with a single condition and later add another condition by clicking on a $\oplus$ button that would expand the block to include an additional condition, and combine the two with a

Boolean operator. For simplicity, in our prototype, we have restricted the block to two conditions at most. Filter c-blocks can be nested if required, offering an alternative way of specifying additional predicates for filtering.

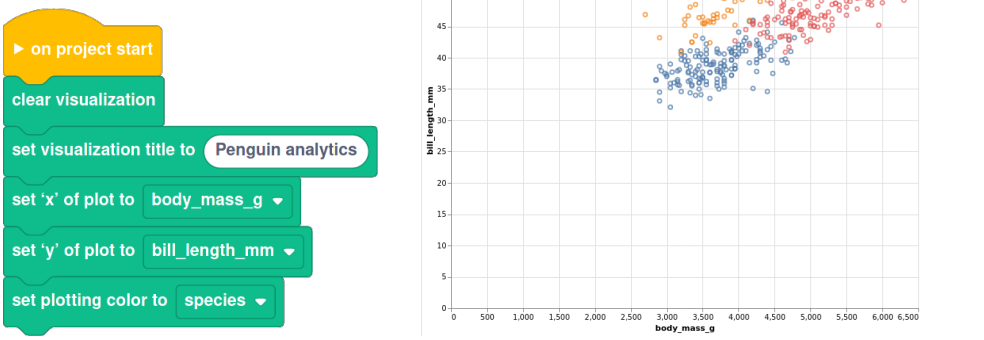
Aggregation

Aggregation, or grouping, is another common task associated with data analysis. We follow a model similar to filtering as described above. A c-block (Figure 6.3b A) allows the *Dataland* user to specify the variable to group by and also the variable to apply the aggregate function on. Aggregate functions, including `count`, `count unique`, `maximum`, `minimum`, `mean`, `median`, `mode`, and `sum`, can be chosen from the drop-down menu. Inside the c-block, code blocks can access the group and the corresponding aggregation result. For example, Figure 6.1a shows how the `group by` block may be used to create a visualization of grouped data. As with the `filter` block, the visual representation of the data table is updated to show the grouped data when the `group by` block is executing (Figure 6.3b B), and an "ungroup" label is added to the bottom of the c-block to indicate that the operation is temporary.

Visualization

Data visualization is one of the major ways for data analysts to communicate their findings and conclusions, and it is also a crucial tool for exploring datasets and finding potential relations among variables or data columns. *Dataland* supports a set of "canonical plots": the standard and commonly used types of Cartesian plots that include scatter plots, line plots, and bar plots. To design visualization code blocks, we followed Wilkinson's grammar of graphics (Wilkinson, 2005) and *Vega-Lite* (Satyanarayan et al., 2017) specifications closely (Figure 6.4) to have atomic functions, with fewer blanks for users to fill in. We have also simplified Wilkinson's plotting process by hiding and automating certain steps (Wilkinson, 2005, p. 39). Here again, we are working with two of the design principles suggested by Resnick and Silverman: that we should keep the floor low for novices, while finding the simplest way to do the most complex things (Resnick and Silverman, 2005, p. 119). Figure 6.4a demonstrates how columns in a dataset (attributes of the data) are coded for visualization, and Figure 6.4b shows the result.

In addition to blocks for canonical data visualizations, *Dataland* also supports plotting on geographical



(a) Sample visualization code. **(b)** Result of running sample code.

Figure 6.4: Visualizing data with *Dataland*

maps. The process is similar, but partly due to technological constraints and partly to keep the number of blocks being presented to learners minimal, blocks for map-based visualizations are available in a separate version of the editor.

6.3.2 Storyline

Computational projects by young people (e.g., in Scratch) are often in the genre of games, animations, interactive stories, etc. Works in these genres usually stand alone in our culture (e.g., we come across games as standalone cultural objects), whereas computational data analysis projects usually are encountered within a broader context (e.g., as a part of a newspaper report). This prompted us to consider the importance of embedding data analysis projects within story-like broader contexts. As a result, a computational notebook-inspired (Kluyver et al., 2016) story-based interface (referred as "storyline") was created for users of *Dataland* to research and analyze data to find answers (Figure 6.1b). Following the storyline, we expected that learners will explore data, gain insights by filtering and cleaning data, produce data visualizations, and interpret their findings—all essential skills for building data literacy.

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

to our research setting, for example, by incorporating references to neighborhoods and landmarks in the city where we conducted our workshops.

Penguin A and B were developed around the tale of a traveling penguin who is searching for their missing cousin after a snowstorm, while Poodle centers on a young boy and his missing dog. For both stories, the learner using *Dataland* would take on the role of a researcher assisting in the search. Penguin A and B are based on the Palmer Islands penguin dataset (Horst et al., 2020), and Poodle is based on a public-domain children's storybook by Mabel Stryker (Stryker, 1923) and uses data sourced from a combination of Seattle and Austin open data portals. For all the stories, we had to add some synthetic data (e.g., new columns) to the original datasets to incorporate the computational data literacy concepts that we aimed to cover.

Each story was structured as a set of interconnected puzzles, with each puzzle being a data analysis task that could be done with a *Dataland* program. For designing the tasks, we drew inspiration from *Sahaj Path*—a primer for the Bengali language, written by Rabindranath Tagore (Tagore, 1930). Each short story or poem in *Sahaj Path* emphasizes a specific alphabet, and we followed this approach by designing each task to focus on one or two data literacy concepts (e.g., filtering). The text of our storylines can be found in the supplemental material of this paper.

6.4 User studies: Dataland Workshops

6.4.1 Workshop and participants

We ran a total of 7 research workshops in 2022 with 27 participants (referred to as P1 to P27) in total. There were 9, 7, 1, 4, 2, 3, 4 participants who participated in the workshops on May 14th, May 15th, June 4th, June 5th, November 9th, November 12th, and November 13th, respectively. Most of the participants only attended one workshop, with the exception of P9, P3, and P12. P9 attended both of the May workshops that used the Penguin A storyline, and P3 and P12 who each attended a May workshop and a June workshop. We used the Penguin A storyline (see §6.3.2) in the first 2 workshops in May (with P1, P3, P5 - P17), the Penguin B storyline in 2 workshops in June (with P2 - P4, P12, and P18), and the Poodle storyline in the remaining 3 workshops in November (with P19 - P27).

All the workshops were hosted on a weekend morning or a weekday evening at our institute and lasted

around 3.5 hours. During the workshop, participants were instructed to interact with the *Dataland* system to follow a storyline to conduct data analysis. Researchers in our team facilitated the workshops, taking field notes, and asking questions to participants. At the end of each workshop, participants were given the option to participate in an interview, where they reflected on their experience using *Dataland* and working with data. Details on the profile of our participants, how we selected the participants, and how we engaged with the participants in the workshops can be found in §6.5.

6.5 Selection and Participation of Children

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

6.5.1 Information about the participants and recruitment

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

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

6.5.2 Participation in Dataland workshops.

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

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

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

6.5.3 Data and Analysis

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

We followed a thematic analysis procedure (Braun and Clarke, 2006) to analyze field notes and interview data. Three researchers who had facilitated workshops first independently annotated lines of interview

transcripts and field notes with notes for possible themes. This round of coding was guided by the structure of the Computational Thinking framework by Brennan and Resnick (2012a) and focused on constructing themes that fell in the categories of "concepts", "practices", and "perspectives". The research team then discussed the codes, identified common themes, reached a consensus on the codes, and collaboratively constructed a codebook. The researchers then reconducted two additional rounds of coding to iteratively merge and synthesize a final set of themes, which are presented in the following section.

6.6 Findings

In this section we report our findings as a framework for studying and developing computational data literacy, following the categorical structure of Computational Thinking proposed by Brennan and Resnick (2012a)—concepts, practices, and perspectives. At a fundamental level, participants in our studies were writing computer programs in a language that shared some characteristics with Scratch, and hence engaged with at least a subset of Brennan and Resnick's concepts, practices, and perspectives. However, in this section, we focus on aspects of these dimensions that go beyond what has already been described by Brennan and Resnick and are more specific to computational data literacy. Building on evidence from our *Dataland* workshops, our framework describes the new **concepts (CO)**, **practices (PR)**, and **perspectives (PE)** that learners can learn through programming with data, and we list and define them as follows.

6.6.1 Concepts

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

CO1: Filtering

One of the most important concepts that participants learned about using *Dataland* is *filtering*—narrowing down a given dataset based on one or more conditions. Following a given data story, participants applied the

`filter` block on the dataset to make visualizations and solve problems. In this process, participants were able to reach a practical understanding of how filtering data works.

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

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

Additionally, being able to see in real time how the execution of the program would impact the data table helped participants understand filtering. For example, P21 made a program that contained a `filter` block. In order to figure out how filtering worked, P21 looked at the data table while the program was being executed. They[3] observed that once the program entered the filter, the data table will switch to the filtered version that contained fewer rows, and switched back once the execution finished. They thus realized that the code inside a `filter` block ran on the filtered dataset.

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

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

CO2: Aggregation

Another important and complex concept that *Dataland* offers to teach was *aggregation*—grouping data based on certain categorical attributes and then performing operations on the groups. At first, many participants did not understand what the `group by` block was and were confused about what the output of aggregation would look like. For example, based on the name of the block, some participants initially

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

guessed that it would help them sort data into groups: "I thought it would like put things into categories. But I don't think it does that." (P19) Through playing with the `group by` block and plotting data, participants were able to gradually realize key underlying concepts. Participants explained aggregation in their own words:

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

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

CO3: Variables

Participants also learned to use the concept of variable to store results and facilitate operations on data in our workshop[4]. One salient example was the storyline tasks that ask users to count the number of items that fit certain conditions. To do this task, participants needed to understand how to create a variable and use the `set variable` block to assign a value to it. Most participants were able to successfully learn to count using variables and understand the concept of variables despite initial challenges.

For example, P12 made a program that used the `filter` block, the `number of rows` block, and variables to count the number of penguins that fit certain conditions in the Penguin dataset. They explained the process of using variables to count data as: "the set block sets the variable that I made to whatever numbers it is, in this case is the number of rows of Adelie [penguins] and those smaller than 4500 grams."

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

In another instance, P4 made a loop to solve a similar counting problem and in this process, they learned about variable initialization and increment: they first created a variable called `Adelie_count`, set it to 0 using the `set variable` block, then used the `repeat` block to create a loop. P4 explained this program that they created as "each time it increases, I will increase `Adelie_count` by one."

In cases like these, participants not only learned about assigning values to variables, but also understood the nature of variables as a place to store values that could dynamically change. An important aspect to note here is that the `repeat`-based approach adopted by P4 does use the `number of rows` block as input for `repeat`, so P4 was certainly aware of the existence of that particular block. Even then, P4 added what was essentially redundant code to count the rows by incrementing the variable. This suggests that, for learners to understand the idea of operating on entire datasets, (described in §6.6.2) instructors may need to provide more active support.

CO4: Statistical concepts

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

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

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

CO5. Visualizations

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

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

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

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

6.6.2 Practices

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

PR1: Conducting operations on entire datasets

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

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

PR2: Processing data and using the results

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

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

the blanks" to clean up the missing values as part of data processing.

PR3: Tinkering with data and code

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

PR4: Cross-checking and debugging with data

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

PR5: Iterating on data presentation

We observed that participants were able to iteratively fine-tune the presentation of data to effectively communicate ideas. They would adjust their visualizations to better represent their data, such as selecting which

part of the data being plotted.

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

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

6.6.3 Perspectives

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

PE1: Large datasets are incomprehensible without computational support

Participants were able to understand why computational methods are necessary with large datasets. *Dataland* offers users the opportunity to think about how to approach and work with large datasets. All but one participant in our workshop had never worked with data on a scale of thousands of rows and multiple dimensions. When seeing the data table for the first time at the beginning of the workshop, it was common for participants to be surprised by the scale of the dataset. Many participants would scroll through the data table in the hope of getting to the bottom and seeing how many rows it contained. In other cases, both of P22 and P26's first action towards the Missing Animal dataset was to use Ctrl + F to find the keyword "Poodle," before realizing that the count might not work as the dataset might be too large to render: "there will be just like too much to count the actual things by hand!"[5]

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

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

PE2: Data is shaped by human decisions

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

PE3: Data can be incomplete and "messy"

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

PE4: Outliers can impact visualization and analysis

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

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

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

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

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

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

6.7 Discussion

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

6.7.1 A framework for studying and developing computational data literacy

In §6.6, we build on Brennan and Resnick (2012a) and describe a framework for computational data literacy that involves understanding new concepts, adopting new practices, and acquiring new perspectives through programming with data. It is crucial to clarify here that we do not claim that programming is the only pathway to these concepts, practices, and perspectives. Rather, our findings demonstrate how scaffolding learners to program with data can facilitate the acquisition of these elements. Furthermore, many aspects of our framework are not merely new additions, but ones that may require a shift from pre-existing knowledge. For example, **PR1** represents a shift from using a combination of conditionals and loops—things that operate on individual data points—to procedures that operate at the level of a collection of data points. We hypothesize that this phenomenon of the need to shift from pre-existing CT knowledge will also be evident in aspects of computational literacy that we have not explored in this paper, such as the concept of vector operations with data. Additionally, the concepts that we surfaced in §6.6.1 overlap with and complement the list of competencies for children to reason with data outlined in Rubin (2020). As several of this broader

set of concepts are traditionally recognized as key and challenging in data science education, our findings indicate that children are able to construct their own understandings of those concepts with a system like *Dataland*. Several practices that we observed in our workshops also echo the exploratory and iterative workflow of professional data analysts (Kandel et al., 2012; Crisan et al., 2020), and our findings provide an unique perspective on how children can organically build up those practices off their preexisting knowledge on programming and data. Furthermore, several perspectives that we observed from our workshops speak to the calls for promoting critical data literacy among children (Lee et al., 2022). In particular, **PE2**, **PE3**, and **PE4** address known perceptual gaps about data that exist among children, such as whose and what decisions are involved in data analysis (Wang et al., 2022).

That said, we do not claim that these represent a comprehensive view—our findings are ultimately mediated and limited by our tool and pedagogical approach, such as limiting data structures to simple scalar variables, or not supporting remixing, which has been studied as a key practice for computational data literacy (Yalcinkaya et al., 2022). Similar limits also apply to the original framework by Brennan and Resnick (2012a), who were largely informed by empirical evidence from the Scratch programming language and online community. We offer these additions to the framework as a starting point for scholars and practitioners interested in computational data literacy, with the hope that new tools and pedagogical approaches will develop the framework even further.

6.7.2 Design implications for data programming systems for youth

Building on advantages of block-based editing, but for data

The design of data programming systems should support its user to easily program and tinker with data. This echos the design principle of "low floor" (Resnick and Silverman, 2005). Our design of *Dataland* offers an example of this principle in action. While it is already known that the visual block-based language lowers the burden for learners to remember the syntax of a programming language (Bau et al., 2017), for data in particular, fixed drop-down menus with data column titles further lowered the burden of memorizing and typing names. The shapes and snapping interaction of visual blocks also provides hints on the compatibility of different data operations. Furthermore, in *Dataland*, learners can easily see the immediate result of their analysis programs to identify and fix issues within the process. All these designs were crucial for learners

to construct their own understanding of unfamiliar data concepts through tinkering and experiments.

Choosing and using black boxes carefully and intentionally

Building on Resnick and Silverman (2005)'s principle of "choosing black boxes carefully,", we saw it play out in specific ways for programming with data. For example, we explicitly wanted learners to know about filtering, and hence included a `filter` block, rather than making learners implement equivalent functionality through loops and conditionals. This type of design choice also echos the design principle that learner-centered data analytic systems should offer learning-centered scaffolds on specific concepts (D'Ignazio and Bhargava, 2016). However, we did see instances where learners considered the use of conditional blocks (§6.6.2) versus the `filter` block. This suggests that the question of black boxes applies not just to the design phase, but also to the learning or use phase, and it is possible to have a useful discussion of when it might be useful for the learner to rely on the black box, vs. when opening up the black box might make sense.

Guidance and creativity

Finally, the design of data programming systems for young people should provide learners with guidance on how to approach data while leaving enough space for creative explorations. Without adequate support, novices find it difficult to explore a large dataset in a meaningful way (Beth Kery and Myers, 2017; Wongsuphasawat et al., 2019). Echoing prior design knowledge about supporting children to inquire with data and expand their inquiries (Wolff et al., 2019), in the design of *Dataland*, we guided learners through a series of questions in storylines and at the same time, we left the specific solutions open for participants to figure out. Most participants found this arrangement to be effective as they did not have to start the analysis from scratch while still having the agency to figure out specific implementations. But some participants also commented that *Dataland* reminded them of school assignments where they simply followed the instructions and had limited space for innovation, citing that they could not develop their own questions and analysis plan with the data. This type of functional fixation is a limitation in our design, and echos other studies on children's learning of computational concepts in constructionist systems (Cheng et al., 2022b). Another limitation of our paper is that our study does not fully address the broader social and cultural contexts of data and data

literacy, as called for by Lee et al. (2022) on humanistic approaches to data science education. We call for future designs to consider these issues and explore ways to provide learners with sufficient guidance while encouraging them to form and answer questions with data in their own contexts.

6.8 Summary

This study delves deeply into the concepts, practices, and perspective that learners can develop through programming with data. At a high level, the constructionist environment removes barriers to entry and allows users to experiment with programming concepts around data based on their own prior knowledge and personal preferences, substantiating epistemological pluralism. In particular, while this research began with the goal of supporting novices who have limited experience in programming and statistics to create programs to analyze and visualize data, my findings include an array of competencies and learning pathways that stem from, but also influence, technical abilities. For example, being able to operate data through interacting with visual blocks makes learners reflect on the human decisions hidden in the data collection process; such critical reflection in turn spurs exploration of statistical concepts such as outliers. The concepts, practices, and perspectives supported by the constructionist environment appear to be deeply rooted in, emerging from, and intricately intertwined with one another. To future support the diverse competencies and pathways, a future direction of this work is to provide learners with the ability to construct the learning environment and their own epistemological pathways. This can include allowing learners to customize the toolkits of programming blocks to suit their diverse needs for engaging with data (as observed in our workshops, some participants would like to modify the blocks to match their own understanding and actions). Another direction is to enable learners to create their own storylines, allowing them to ask and answer questions with data that is directly relevant to their own communities.

Chapter 7

Discussion: A Sociotechnical Lens to the Pluralism in Data Literacies

Through the four studies, this dissertation contributes new understandings to previous work on data literacy, especially regarding informal settings. Together, my series of four studies strive to understand what data literacy means to the public's engagement with data in informal settings. At a high level, these four studies broaden our understanding of data literacy, presenting a set of distinct competencies and engagement practices in informal contexts. Together, these four studies form the foundation for my development of a new sociotechnical framework (elaborated in §7.2) that can be used by future researchers to study and design for data literacies in informal settings.

Specifically, Study A focuses on the informal online settings where novices engage with data to enhance their creative projects. In the case of the Scratch online community, for instance, members incorporate data into their creative projects that stemmed from personal interests and subsequently shared with the community. The study A expanded the previous understanding of computational concepts related to data by illustrating what they look like and how they are developed and shaped in an informal online setting. This study recognized significant skills and practices they developed during this engagement: the ability to build an understanding of programming concepts related to data in terms of concrete use cases that could benefit their projects, such as employing data as a score counter or leaderboard in the games they are making. Equally crucial is their ability to articulate specific data needs for their projects to the community and the

ability to search for, evaluate, and apply tutorials created by fellow community members to their use cases.

Study B focuses on the online informal settings where individuals, spanning a range of experience levels in data science, collaborate to analyze and solve problems with large datasets. This study in particular illustrated a body of competencies and practices that were previously overlooked by the literature on data literacy but are crucial for engagement with data in informal settings. As shown by the specific case of Kaggle, an important ability that emerges from this setting is to identify and recruit suitable collaborators possessing the right data skills. Moreover, members cultivate the ability to communicate the rationales and exploratory procedures of data work to various audiences within the informal environment. This includes not only immediate collaborators during a project, but also a broader community when seeking feedback, discussing ideas, and post-project for sharing and educational purposes. This communication is also not merely unidirectional or solely for demonstration; it also encompasses the skills to inquire about and understand others' data procedures.

Study C delves into online informal settings where individuals use data to articulate opinions in public debates surrounding real-world civic issues. Study C contributes to nuanced but important aspects regarding critical data literacy in online informal settings. Specifically examining discussions related to COVID-19 vaccine data on Twitter, the study suggests the importance of the ability to critically evaluate data arguments. This involves recognizing the authors and audiences of data arguments, thinking through the construction, narrative, and meta-delivery of the arguments, and making informed decisions about the credibility of these data arguments.

Study D presents an idealistic setting where barriers to programming with data are eliminated, allowing novices to computationally engage with data by creating computer programs. By offering such a setup, Study D established a detailed framework of the concepts, practices, and perspectives related to data programming, detailing how computational data literacy looks like for novices in informal settings. Specifically, Dataland offers a constructionist environment where even those unfamiliar with computer programming can analyze and visualize data to solve problems. Dataland supports an array of competencies and practices, not limiting to data science concepts, but also encompassing a suite of practices and perspective on learning with and about data.

Beyond the enrichment and expansion of currently understanding of data literacy with the empirical in-

sights from online informal settings, in this section, I would like to discuss the implication of this dissertation on studying and designing for data literacy in informal settings. Specifically:

- **A recognition of pluralism of data literacies in informal settings**. This dissertation calls for recognition of the pluralism that accounts for the different competencies that people should seek when engaging with data across various contexts and communities, as well as the diverse epistemological pathways to get to the competencies. The design of tools and interventions aimed at enhancing data literacies should acknowledge and embrace this pluralism.
- **A socio-technical framework for studying and designing to support data literacies**. This dissertation posits that efforts to understand and support data literacies in informal settings should be approached through a socio-technical lens. This entails a multi-step framework that future researchers of data literacies could follow: starting with recognition of the particular data engagement and relevant platform features, understanding the domain, community, and practices of the community of users in any given setting, delving into the competencies and pathways that shaped by this setting, and deriving underlining challenges and design opportunities.

7.1 Recognizing the pluralism in informal engagement with data

I initially started this dissertation journey with the aim of seeking a clear and centralized understanding of data literacy in informal settings, which in my imagination would be a taxonomy delineating various types of data literacy within informal settings that would serve as a guide for future tools and interventions designs. My initial goal was to create a comprehensive map detailing all concepts, skills, competencies, practices, and perspectives related to informal engagement in the data and to categorize these findings according to epistemological differences and design principles. For example, I initially attempted to label elements related to programming as "technical data literacy", along with anything related to collaboration termed "discursive data literacy", and aspects linked to a wider social context designated "critical data literacy."

However, this approach seemed inadequate to capture the plurality evident in the informal settings that I explored. This plurality, illuminated by my four studies, manifests itself in various forms. It is reflected in the different environments and the corresponding ways of engagement that each setting offers. For instance,

Scratch serves as a platform for novices, primarily children, who pursue interest-driven creations using computational (including data) concepts; Kaggle functions as a platform for individuals with varied experience in working with data to collaboratively, yet competitively, solve problems using large datasets; On Twitter, participants engage in online discussions, leveraging data to backup their arguments and opinions on social civic issues; Dataland, our own platform, presents specific concepts related to data analysis and visualization in the form of building blocks, allowing novices computationally ask and answer with data. These four settings also offer only a glimpse into the myriad of informal settings in the world where people actively engage with data.

7.1.1 From data literacy to data literacies

Given this evident plurality, striving for a centralized understanding of data literacy that reconciles all settings feels unsatisfying and counterproductive. For instance, a significant skill for users of platforms like Scratch and Dataland is the ability to construct knowledge about programming with data in a way that speaks to their creative interests. This skill might not be relevant to people who only consume and are influenced by data-driven arguments on Twitter. Similarly, Kaggle users face the challenge of navigating collaborations and communications around working with large datasets within a hierarchy structure of experience levels. This challenge would be irrelevant in the Scratch online community, where such a ranking structure is absent.

Another aspect of this plurality lies in the interconnections between the unique concepts, skills, practices, and perspectives within a single setting. For instance, in Kaggle, the advancement of programming and statistical skills for handling large datasets are intertwined with the ability to effectively identify and communicate with collaborators and the broader community. Similarly, Dataland was designed to facilitate novices in programming with tabular datasets. My study revealed that the skills to programming with data actually grounded a broader spectrum of practices and perspectives that include critically reflecting on human decisions in the construction of datasets and innovative interpretation of outliers. It becomes challenging to isolate specific competencies and categorize them as either "technical" or "critical" or any other label. Every "literacy" is underpinned by, evolves from, and is intricately linked with a multitude of other "literacies" inherent in a given setting.

My research underscores the multifaceted nature of motivations, skills, practices, and challenges people face when they engage with data in diverse informal settings. Looking more broadly in the literacy education community, there is a growing consensus to shift from a monolithic understanding of "literacy" to a broader, more inclusive notion of "literacies" (Collins, 1995). This nuanced perspective embraces the cultural and historical backgrounds, as well as the myriad learning pathways associated with reading and writing skills that are often overshadowed by mainstream education, but remain important for specific communities.

In alignment with this enriched viewpoint, I argue that the public's engagement with data in informal environments embodies a spectrum of literacies that people are demonstrating and seeking for. It is clear that there is no universal solutions for creating interventions to foster data literacy. When creating technology for data literacies in informal settings, it is important to account for the backgrounds and needs of the intended users, the nature of their practices around engaging with data, the infrastructure of the platform, and the culture of the community.

7.1.2 Epistemological pluralism in data literacies

The pluralism is also illustrated in the varied pathways that individuals take to achieve the diverse competencies around data engagement. This aspect of pluralism speaks to the emphasis of epistemological pluralism in constructionism (explained in Chapter 2), which recognizes that there are multiple ways of knowing and understanding. Learners can construct knowledge in various ways on the basis of their experiences and contexts.

In Scratch, for instance, the understanding of technical concepts related to data is constructed from a bottom-up manner, through examples and use cases that align with learners' interests. Dataland is designed based on a similar philosophy, offering a plethora of methods through which novices can interact with specific data concepts. For example, in Dataland workshops, when participants were introduced to the concept of slicing data, some utilized the filtering function block we designed, while others employed conditions and loops to achieve the same result. Numerous such instances underscore that constructionist informal settings allow individuals to develop concepts and skills around data following their unique interests, prior knowledge, and objectives.

Even on platforms not explicitly designed based on constructionist principles, my studies were able to

reveal diverse pathways of data engagement. On Kaggle, community members adapt their data analysis communication to cater to a diverse audience with distinct needs and in different contexts. This includes customizing the result presentations for various stakeholders: direct collaborators, potential competitors, and the broader learning community. On Twitter, individuals exhibit a variety of strategies to engage with visualizations and datasets, and employ a variety of rhetorical techniques to critically interact with data concerning social issues.

The shift from discussing "data literacy" in the singular to "data literacies" in the plural is more than just a semantic nuance. Rather, I argue for its importance as a critical expansion of the way we understand how people engage with data in informal settings. Although previous approaches to data literacy often imply a one-size-fits-all skill set and method that individuals should follow to make sense of data, this dissertation provides viewpoints on the diverse contexts, purposes, and sociotechnical settings in which people engage with data. At a high level, this dissertation brings up a framing of discussions around data literacies in the plural, with the purpose to call for attention on the pluralism that accounts for the different abilities, skills, practices, and competencies that people should seek for when engaging with data across various contexts and communities.

7.1.3 Implications for tools and interventions to support data literacies in informal settings

The pluralistic nature of data literacies offers a new lens of design opportunities. Recognizing the diverse ways people engage with data suggests that specialized tools and interventions, tailored to distinct user needs and scenarios, are crucial. My four studies present a range of design considerations for informal environments that support data engagement, emphasizing diverse competencies and pathways related to data literacies.

Study A indicates that informal learning platforms should mitigate the effect of the social feedback loop that I identified and ensure that users can learn data concepts from diverse interests. Plaform affordances and regulation mechanisms could aim to support a wider range of inspiration, broaden the participation of members with diverse interests, and scaffold users to connect the knowledge that they are building to generalizable concepts.

Study B underscores the importance of ensuring the collaborative work of users of different backgrounds

and levels of experience when working with large datasets. To achieve this, it is crucial to address the disconnection between members on the opposite ends of the spectrum of experience levels. The features and policies of the platform could promote beginner-friendly presentations of data analysis procedures by experts, ensuring thoroughness and comprehensibility. Furthermore, platforms could implement mechanisms to reduce barriers for beginner engagement and enhance their interaction with experts, such as promoting the visibility of beginner's contribution while protecting their vulnerable status, as well as incorporating team-formation strategies to pair beginners with experts in safe spaces.

Study C calls for the promotion and protection of critical data literacy in informal contexts. Platforms could facilitate users not only to critically argue with data but also to critically reflect on the arguments, such as assessing the narrative structure and delivery of arguments. Design considerations could include scaffolds that direct attention to rhetorical characteristics, such as causal claims, certainty, and references to authority highlighted in this study, prompting users to identify conspiracy claims and be mindful of their arguments with data.

Study D presents insights for designing systems that support novice to program with data. For people with limited programming skills to work with data, a constructionist design approach can remove barriers to entry and allow them to computationally make sense of data based on their own prior knowledge and personal preferences. Future systems can further support learners to construct the learning environment and their own epistemological pathways, such as allowing learners to customize the constructionist toolkits to suit their diverse needs of engaging with data, and to enable learners to ask and answer questions with data that is directly relevant to their own communities.

7.2 A sociotechnical framework to studying and designing for data literacies

My research demonstrates the multiple competencies and epistemological pathways regarding informal data engagement shaped by the various features of informal settings and the diverse characteristics and needs of user communities. As researchers designing tools and interventions to support data literacies, we must honor this diversity in the *problem space* of data literacies, that is, what exact *kind* of data literacies should we design for? My studies included in this dissertation indicate that data literacies in informal settings are situated in specific contexts, which consist of particular forms of data engagement, specific platform

affordances, and the structures and values in user community. As a result, understanding and unpacking the complexity of these aspects is crucial in the design of technology and research to support data literacies.

To make a systematic approach to these questions, I conclude my dissertation by introducing a framework tailored to analyze and design for data literacies within informal contexts. This framework delineates four important steps for studying and designing for data literacies in a given informal setting. The framework is presented in Figure 7.1. I will elucidate this framework in subsequent sections and employ it to my four studies as examples of how to use it to investigate data literacies.

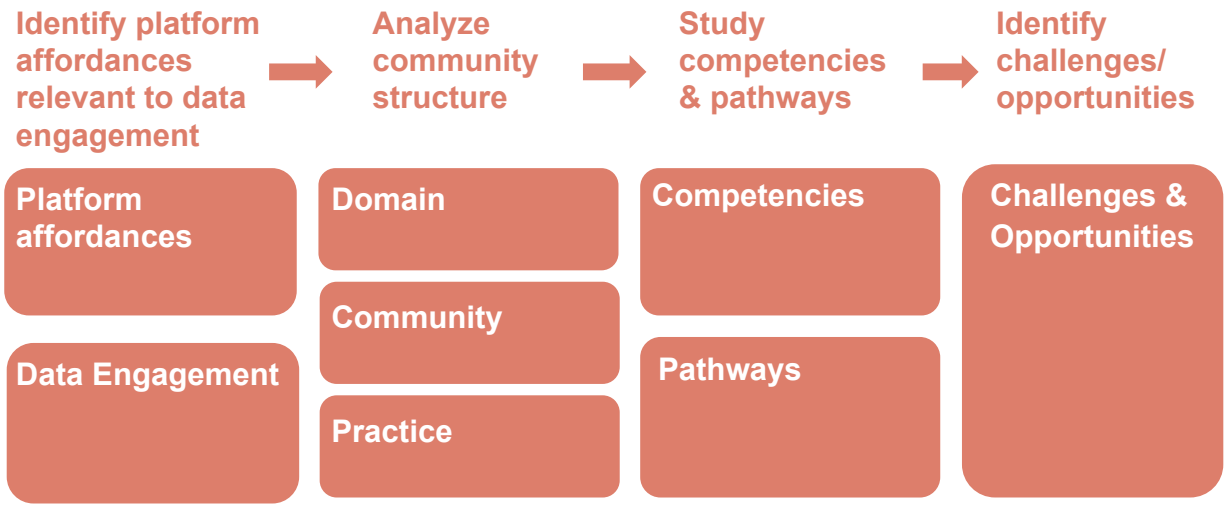


Figure 7.1: The sociotechnical framework to studying and designing for data literacies

Step 1: Identify platform affordances relevant to data engagement.

This initial step denotes the preliminary efforts of the investigation. Researchers should choose an informal setting of interest and determine the nature of data engagement they are aiming to explore. This requires defining the specific **data engagement** and identifying relevant **platform features**. For instance, within Scratch, data engagement refers to users using data to create creative programming projects, while the relevent platfrom features include the parts of the block-based programming system that involves data, functionalities to share code through remixing and downloading projects, and the text-based online discus-

sion where users interact with each other around projects. In platforms that are not explicitly designed for data engagement (e.g., Twitter), prior observations and literature can guide its identification. For example, in study C on Twitter, insights from Lee et al. (2021) and other literature on vaccine discourse revealed that users frequently discuss COVID-19 vaccines, using data and visualizations to substantiate their views. This data engagement is supported by Twitter's online discussion features that combines text and media (image, video, link to external sources), as well as social features to share and interact with posts.

Step 2: Analyze community structure based on platform affordances

Following the identification of platform affordances relevant to data engagement, Step 2 requires an analysis of the user community and its values. By adopting the Community of Practice structure delineated in Chapter 2, the user community can be dissected into three components: domain, community, and practice. The previously recognized platform features and data engagement can inform this analysis. The **domain** indicates users' primary objectives within the context of data engagement. For instance, within Study A's context of Scratch, it is to understand the structural and functional uses of variables and lists as the underlining concepts of data blocks. The **community** reflects the identity and characteristics of the user involved in the particular data engagement. For Study A, this involves primarily novice users (predominantly children) making, sharing, and interacting around creative programming projects. The **practice** encompasses user activities and the shared culture afforded by the features of the platform. In Study A's context, this translates to the shared practice of making and publicly sharing projects with data blocks and asking and answering questions around them.

Step 3: Study competencies and pathways regarding community structure

After recognizing the community structure afforded by the platform, the next phase involves exploring the array of data literacies situated within this specific environment. Specifically, we can look at the **competencies** and **pathways** that users develop or seek to develop with respect to the domain, community, and practice of the sociotechnical system. Competencies encapsulate the skills, concepts, and perspectives that users cultivate. For example, in Study A, a competency regarding the domain involves learning the concept of variables and lists; a competency regarding the community involves identifying relevant community resources; and a competency regarding the practice involves using variables and lists to support interest. Pathways represent the mechanisms that guide users towards the competencies. For instance, in Study A, a

pathway towards the domain involves making game elements with data blocks; a pathway towards the community involves asking and getting help from other novices; and a pathway towards the practice involves building and using community learning resources with popular use cases in games. Although it may not be interesting to analyze competencies and pathways related to all three aspects of domain, community, and practice, this framework hopes to provide a structure for identifying relevant competencies and pathways.

Step 4: Identifying challenges/opportunities

Upon discerning the competencies and pathways that comprise data literacies, the final step involves identifying challenges and opportunities by analyzing potential gaps or frictions between the competencies and pathways, and all previous elements of the framework. Typical questions that can be asked may include the following: do users encounter obstacles in their journey towards competencies? Are certain user groups more predisposed to challenges in skill acquisition? Do specific platform features hinder certain pathways? Insights into challenges and potential design implications can be gleaned from this framework. For example, in the case of Study A, the social feedback loop that narrows the use cases of data concepts in the community represents the gap between the pathway of building and using community learning resources with popular use cases in games, and the competency of using variables and lists to support the users' own personal interest.

In particular, this framework presents a novel approach for researchers to identify specific data literacies they are designing for and to uncover design opportunities. In addition, each step within the framework also contributes uniquely to understanding the context of data literacies. This versatility allows the framework to be applied at various stages of research and design, whether it is analyzing and iterating on the features of an existing system to promote data literacies, or in the conceptualization and planning of new features or new systems. This adaptability makes the framework effective both in the evaluation and in the creation of systems for data literacies.

I applied this framework in my four studies (see figures 7.2 to 7.5). This exercise showcases how data literacies in informal settings can be effectively investigated. To avoid redundancy, I will briefly outline the analysis of each study.

Figure 7.2 illustrates the analysis of Study A, with a focus on analyzing an existing system (i.e., Scratch) and unpacking the design opportunities of its existing features in promoting data literacies. Since most of

Figure 7.2: Analysis of Study A using the §7.2 framework

the details were discussed above in the introduction of the four steps, I will not elaborate on them here.

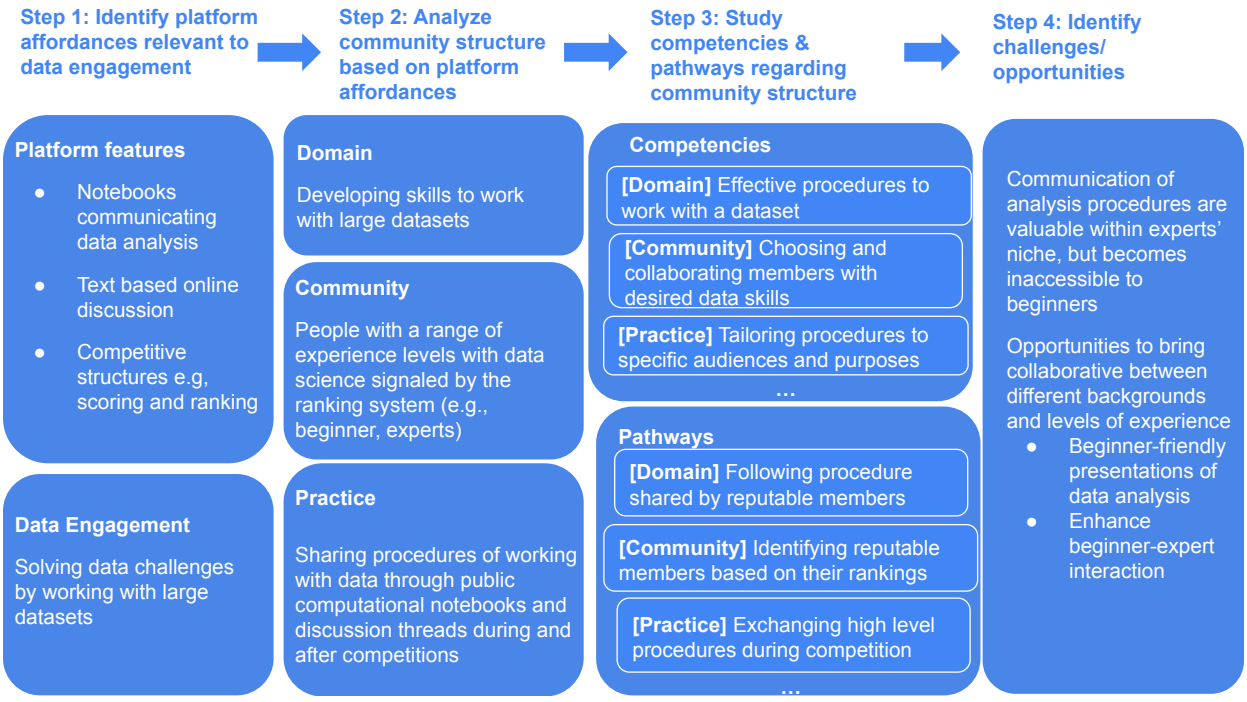


Figure 7.3: Analysis of Study B using the §7.2 framework

Figure 7.3 illustrates the analysis of Study B, also representing the analysis of an existing system and identifying design spaces based on its existing features. During my research, delineating platform features, such as competition structure elements (Step 1), enabled me to pinpoint structures of the user community (Step 2), such as the hierarchy of user experience levels, and the practice of sharing procedures of data analysis during and after competitions. These insights support the identification of the competencies and pathways related to data literacies in this unique setting (Step 3). For example, a competency regarding the domain is to learn new, effective procedures of working with a particular dataset, while pathways include identifying reputable members based on their rankings and following procedure shared by reputable members, and the practice that reputable members tend to exchanging only high-level procedures during competitions. These pathways can lead to friction with the competency, resulting in the challenge where high-level procedures of data analysis shared by reputable members are valuable within their niche but become inaccessible to beginners.

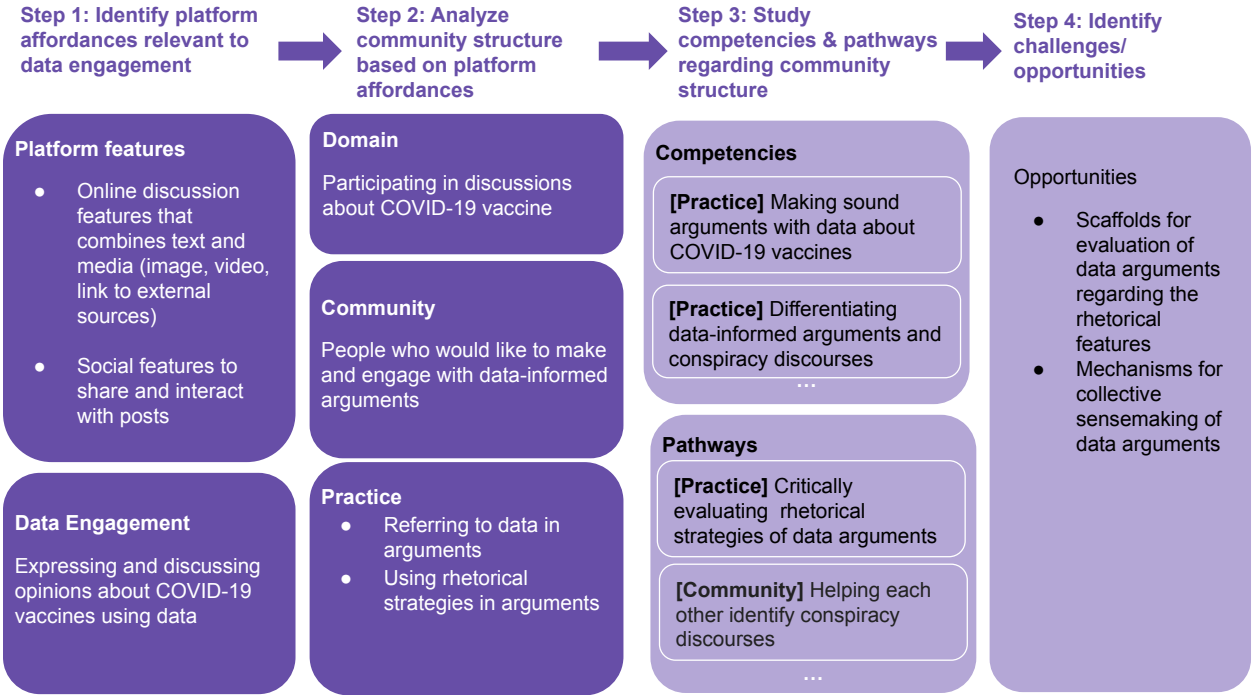


Figure 7.4: Analysis of Study C using the §7.2 framework

This framework can also help researchers design new research and novel features as addition to existing systems. Figure 7.4 depicts this potential through the analysis of Study C, where the darker colored pallets

represent insights from the current study and the lighter colored pallets indicate future opportunities. By connecting platform features and data engagements with previous literature, this study focuses on differentiating critical data engagements from conspiracy claims. The major contribution of the current version of the study is to distinguish data-informed arguments from conspiracy-driven narratives, which is mainly found in Step 2. However, its implications can be inferred from our framework. Based on the domain, community, and practice of the user community, the competencies that people can seek to develop can include making sound arguments with data on COVID-19 vaccines and separating data-informed arguments and conspiracy discourses. Potential pathways that can be designed to foster include critical evaluation of rhetorical strategies of data arguments and helping each other identify conspiracy discourses. This leads to opportunities for platform features and community mechanisms to support critical evaluation of rhetorical strategies for data arguments.

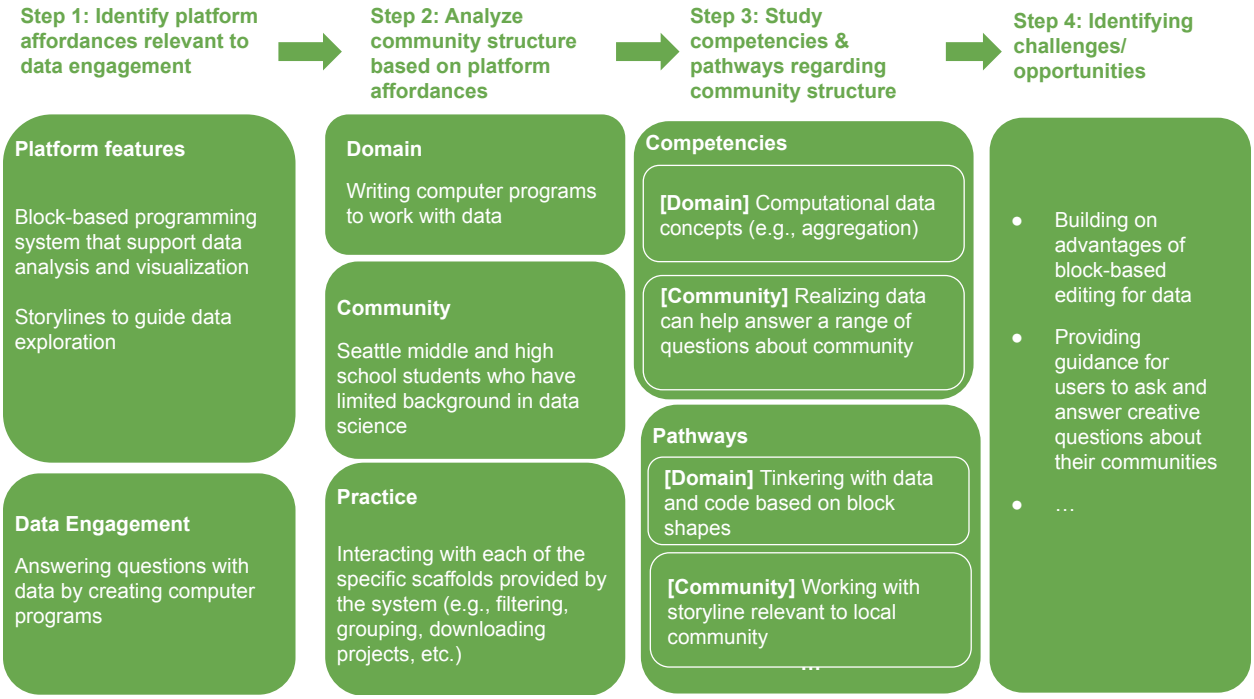


Figure 7.5: Analysis of Study D using the §7.2 framework

Lastly, Figure 7.5 represents the process of conducting Study D, which features the design of a new system and the evaluation of that system regarding the types of data literacies it can promote. The starting goal was to introduce the specific data engagement of novices answering questions with data by creating

computer programs. To support this data engagement, we created the block-based programming system specifically for data analysis and visualization, guided by storylines of a series of connected data problems. The system was aimed for the user community to learn the domain skills of writing computer programs to work with data, targeted at local middle and high school students with a limited background in data science, and to scaffold them to interact with the blocks. By deploying the system in workshops, we observed various competencies and pathways related to data. For example, participants were able to build their understanding of a series of computational data concepts through tinkering with code based on the design of the blocks. They were also able to realize that data can answer a range of questions. By analyzing these competencies and pathways as well as their relationships with the platform affordances we designed, we learned that block-based, story-driven systems could function as an effective approach for novices to program with data, while users need scaffolds and guidance to tell stories about their communities through data.

7.3 Looking forward: Data literacies in the era of generative AI

As the research presented in this dissertation was conducted in the pre-generative AI era, it is imperative to look ahead and consider emerging technologies like generative AI in the landscape of data engagement and literacy. The advent of generative AI promises a future where many aspects of interacting with data could be significantly automated. We have seen the early signs of this revolution in fields like programming, where tools like Copilot are dramatically simplifying the coding process, and in writing, where models like ChatGPT are transforming how we generate text.

Given these rapid, revolutionary advancements, it seems inevitable that the influence of generative AI technology on the public's interaction with data (which I would like to refer to as "AI-influenced data engagement") is on the horizon. One key question that emerges for future exploration is: What does it mean to promote data literacy in an age where automation is increasingly taking over? I have been contemplating this question from two distinct angles.

The first angle considers investigations of the concepts, practices, and perspectives that humans should cultivate when engaging with data, especially when many tasks can be automated by AI. Increasing research has shown that the proliferating AI-powered code generation tools can automate data tasks that are often considered tedious by learners, such as writing code to clean data and build models, removing many obsta-

cles related to technical skills. Perhaps, for learners, the primary focus should now shift towards cultivating a "mindset" for data engagement. This can involve developing the ability to ask meaningful questions and making meaningful plans about data (or having the ability to determine what a good question is). Many of these competencies can potentially align with the data literacies explored in this dissertation—such as determining how to best use data to achieve end-goal or to support creativity, or effectively presenting data-driven arguments.

The second angle involves a novel set of competencies and epistemological pathways to be investigated concerning the AI systems themselves. The framework presented in §7.2 can be used to understand human practices, challenges, and needs in understanding the implications, limitations, and ethical considerations of data outcomes generated by automated systems. For example, outside of this dissertation, my own work on the software developer's usage of GitHub Copilot (Cheng et al., 2023b; Wang et al., 2023) maps a series of insights regarding how users of the code generation AI understand its limitations, prompt and harness the AI, recognize potential biases, and critically assess and make decisions of AI outputs. Future research could aim to understand and contextualize these findings in relation to AI-influenced data engagement.

The implication of this dissertation on studying data literacies from a sociotechnical lens can remain relevant even in the age of AI-influenced data engagement. My research on the GitHub Copilot user community (Cheng et al., 2023b) indicates that user communities of generative AI provide a shared platform for members to understand use cases, devise strategies, understand implications, and evaluate methods related to AI-generated content. As AI takes on more tasks related to data, it will be invaluable to study how communities in informal online spaces adapt, evolve, and influence the ways in which individuals interact with AI-driven output. In the future of AI-influenced data engagement, researchers, educators, policymakers, and AI developers should work together to understand user practices, challenges, and needs in online informal settings and devise novel interventions to support the public's engagement with data.

Chapter 8

Closing

This dissertation dives into the various settings and pathways through which the public informally interacts with data. At a high level, it underscores the pluralistic and sociotechnical lenses for studying and designing such engagements. To reiterate its contribution, empirically, this dissertation enriches our understanding with detailed studies of how the public engages with data in four different exemplar informal settings. Each study, in its essence, offers a spectrum of design guidelines that future technology design in informal settings can adopt to foster a diverse range of data literacies. Theoretically, the dissertation calls the research community to recognize the plural competencies and epistemological pathways that consist of data literacies. To help future researchers and practitioners study and design for data literacies in informal settings, this dissertation offers a sociotechnical framework that can serve as a guide to identify specific data engagements and platform features, analyze community structures, and subsequently derive competencies and pathways of interest and identify challenges and opportunities for design.

As I conclude this dissertation, it is important to remember that the landscape of public data engagement is ever-evolving. It is possible that the specific design recommendations mentioned in each study will become irrelevant due to the emergence of new social and technological modalities. However, this very future reinforces the necessity to approach the investigation and design of data literacies with an open, pluralistic mindset. As society and technology move forward, the pathways for the public to interact with data will multiply, and it is my aspiration that the insights from this dissertation serve as a guiding light in supporting such endeavors.

Bibliography

Diaa Salama Abdelminaam, Fatma Helmy Ismail, Mohamed Taha, Ahmed Taha, Essam H. Houssein, and Ayman Nabil. 2021. CoAID-DEEP: An Optimized Intelligent Framework for Automated Detecting COVID-19 Misleading Information on Twitter. *IEEE Access*, 9:27840–27867.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, El Moatez Billah Nagoudi, Dinesh Pabbi, Kunal Verma, and Rannie Lin. 2021. Mega-COV: A Billion-Scale Dataset of 100+ Languages for COVID-19. ArXiv:2005.06012 [cs].

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008a. Finding High-Quality Content in Social Media. In *Proceedings of the International Conference on Web Search and Web Data Mining - WSDM '08*, page 183, Palo Alto, California, USA. ACM Press.

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008b. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, page 183–194, New York, NY, USA. Association for Computing Machinery.

Dalia Almaghaslah, Abdulrhman Alsayari, Geetha Kandasamy, and Rajalakshimi Vasudevan. 2021. COVID-19 Vaccine Hesitancy among Young Adults in Saudi Arabia: A Cross-Sectional Web-Based Study. *Vaccines*, 9(4):330.

Maria J Antikainen and Heli K Vaataja. 2010. Rewarding in open innovation communities–how to motivate members. *International Journal of Entrepreneurship and Innovation Management*, 11(4):440–456.

Nikolay Archak. 2010. Money, glory and cheap talk: Analyzing strategic behavior of contestants in simultaneous crowdsourcing contests on topcoder.com. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 21–30, New York, NY, USA. Association for Computing Machinery.

Alexandre Ardichvili. 2008. Learning and knowledge sharing in virtual communities of practice: Motivators, barriers, and enablers. *Advances in developing human resources*, 10(4):541–554.

John Ashton. 2021. COVID-19 and the anti-vaxxers. *Journal of the Royal Society of Medicine*, 114(1):42–43. Tex.eprint: https://doi.org/10.1177/0141076820986065.

Junjie Aw, Jun Jie Benjamin Seng, Sharna Si Ying Seah, and Lian Leng Low. 2021. COVID-19 Vaccine Hesitancy—A Scoping Review of Literature in High-Income Countries. *Vaccines*, 9(8):900.

Sasha A Barab and Thomas Duffy. 2000. From practice fields to communities of practice. *Theoretical foundations of learning environments*, 1(1):25–55.

Albert-László Barabási and Réka Albert. 1999. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512.

Michael Barkun. 2013. *A culture of conspiracy: apocalyptic visions in contemporary America*, second edition edition. Number 15 in Comparative studies in religion and society. University of California Press, Berkeley.

David Barr, John Harrison, and Leslie Conery. 2011. Computational Thinking: A Digital Age Skill for Everyone. *Learning & Leading with Technology*, 38(6):20–23.

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

Satabdi Basu, Betsy Disalvo, Daisy Rutstein, Yuning Xu, Jeremy Roschelle, and Nathan Holbert. 2020. The Role of Evidence Centered Design and Participatory Design in a Playful Assessment for Computational

Thinking About Data. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, pages 985–991, New York, NY, USA. Association for Computing Machinery.

David Bau, Jeff Gray, Caitlin Kelleher, Josh Sheldon, and Franklyn Turbak. 2017. Learnable Programming: Blocks and Beyond. *Commun. ACM*, 60(6):72–80.

Barry L Bayus. 2013. Crowdsourcing new product ideas over time: An analysis of the dell ideastorm community. *Management science*, 59(1):226–244.

Yochai Benkler. 2002. Coase's penguin, or, linux and" the nature of the firm". *Yale law journal*, pages 369–446.

Burcu Berikan and Selçuk Özdemir. 2020. Investigating "Problem-Solving With Datasets" as an Implementation of Computational Thinking: A Literature Review. *Journal of Educational Computing Research*, 58(2):502–534.

Francine Berman, Victoria Stodden, Alexander S. Szalay, Rob Rutenbar, Brent Hailpern, Henrik Christensen, Susan Davidson, Deborah Estrin, Michael Franklin, Margaret Martonosi, and Padma Raghavan. 2018. Realizing the potential of data science. 61(4):67–72.

Mary Beth Kery and Brad A. Myers. 2017. Exploring exploratory programming. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 25–29, Raleigh, NC. IEEE.

Rahul Bhargava, Ricardo Kadouaki, Emily Bhargava, Guilherme Castro, and Catherine D'Ignazio. 2016. Data Murals: Using the Arts to Build Data Literacy. *The Journal of Community Informatics*, 12(3).

Pavol Bielik. 2012. Integration and adaptation of motivational factors into software systems. In *Personalized Web-Science, Technologies and Engineering: 11th Spring 2012 PeWe Workshop Modra-Piesok, Slovakia April 1, 2012 Proceedings*, pages 31–32.

Sanat Kumar Bista, Surya Nepal, Nathalie Colineau, and Cecile Paris. 2012. Using gamification in an online community. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 611–618. IEEE.

Elizabeth B Blankenship, Mary Elizabeth Goff, Jinging Yin, Zion Tsz Ho Tse, King-Wa Fu, Hai Liang, Nitin Saroha, and Isaac Chun-Hai Fung. 2018. Sentiment, Contents, and Retweets: A Study of Two Vaccine-Related Twitter Datasets. *The Permanente Journal*, 22(3):17–138.

Paulo Blikstein. 2018. Pre-College Computer Science Education: A Survey of the Field. Technical report, Google LLC, Mountain View, CA.

Kevin J Boudreau and Karim R Lakhani. 2015. "open" disclosure of innovations, incentives and follow-on reuse: Theory on processes of cumulative innovation and a field experiment in computational biology. *Research Policy*, 44(1):4–19.

Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101.

Karen Brennan and Mitchel Resnick. 2012a. New Frameworks for Studying and Assessing the Development of Computational Thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada. AERA.

Karen Brennan and Mitchel Resnick. 2012b. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*, volume 1, page 25.

Karen Brennan and Mitchel Resnick. 2013. Imagining, creating, playing, sharing, reflecting: How online community supports young people as designers of interactive media. In *Emerging technologies for the classroom*, pages 253–268. Springer.

Karen Brennan, Amanda Valverde, Joe Prempeh, Ricarose Roque, and Michelle Chung. 2011. More than code: The significance of social interactions in young people's development as interactive media creators. In *EdMedia+ Innovate Learning*, pages 2147–2156. Association for the Advancement of Computing in Education (AACE).

David A. Broniatowski, Amelia M. Jamison, SiHua Qi, Lulwah AlKulaib, Tao Chen, Adrian Benton, Sandra C. Quinn, and Mark Dredze. 2018. Weaponized Health Communication: Twitter Bots and Russian Trolls Amplify the Vaccine Debate. *American Journal of Public Health*, 108(10):1378–1384.

Amy Bruckman. 1998. Community Support for Constructionist Learning. *Computer Supported Cooperative Work (CSCW)*, 7(1-2):47–86.

Amy Bruckman. 2005. *Learning in Online Communities*, Cambridge Handbooks in Psychology, page 461–472. Cambridge University Press.

Susan L Bryant, Andrea Forte, and Amy Bruckman. 2005. Becoming wikipedian: transformation of participation in a collaborative online encyclopedia. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 1–10.

Leah Buechley and Benjamin Mako Hill. 2010. LilyPad in the Wild: How Hardware's Long Tail Is Supporting New Engineering and Design Communities. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems (DIS '10)*, pages 199–207, New York, New York. ACM Press.

Angelika C Bullinger, Anne-Katrin Neyer, Matthias Rass, and Kathrin M Moeslein. 2010. Community-based innovation contests: Where competition meets cooperation. *Creativity and innovation management*, 19(3):290–303.

Inc Bunchball. 2010. Gamification 101: An introduction to the use of game dynamics to influence behavior. *White paper*, 9.

Moira Burke, Cameron Marlow, and Thomas Lento. 2009. Feed me: motivating newcomer contribution in social network sites. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 945–954.

Talha Burki. 2019. Vaccine misinformation and social media. *The Lancet Digital Health*, 1(6):e258–e259.

Julie Campbell, Cecilia Aragon, Katie Davis, Sarah Evans, Abigail Evans, and David Randall. 2016a. Thousands of positive reviews: Distributed mentoring in online fan communities. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, CSCW '16, page 691–704, New York, NY, USA. Association for Computing Machinery.

Julie Campbell, Cecilia Aragon, Katie Davis, Sarah Evans, Abigail Evans, and David Randall. 2016b. Thousands of Positive Reviews: Distributed Mentoring in Online Fan Communities. In *Proceedings of the 19th*

ACM Conference on Computer-Supported Cooperative Work & Social Computing, CSCW '16, pages 691–704, New York, NY, USA. ACM.

Dana M. Casciotti, Katherine C. Smith, Amy Tsui, and Ann C. Klassen. 2014. Discussions of adolescent sexuality in news media coverage of the HPV vaccine. *Journal of Adolescence*, 37(2):133–143.

Huseyin Cavusoglu, Zhuolun Li, and Ke-Wei Huang. 2015a. Can Gamification Motivate Voluntary Contributions?: The Case of StackOverflow Q&A Community. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing - CSCW'15 Companion*, pages 171–174, Vancouver, BC, Canada. ACM Press.

Huseyin Cavusoglu, Zhuolun Li, and Ke-Wei Huang. 2015b. Can gamification motivate voluntary contributions? the case of stackoverflow q&a community. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing*, CSCW'15 Companion, page 171–174, New York, NY, USA. Association for Computing Machinery.

Lidia Ceriani and Paolo Verme. 2012. The Origins of the Gini Index: Extracts from Variabilità e Mutabilità (1912) by Corrado Gini. *The Journal of Economic Inequality*, 10(3):421–443.

Joel Chan, Steven Dang, and Steven P. Dow. 2016. Comparing Different Sensemaking Approaches for Large-Scale Ideation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2717–2728, San Jose California USA. ACM.

Kathy Charmaz. 2006. *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. Sage Publications, London, UK.

Shuchi Chawla, Jason D. Hartline, and Balasubramanian Sivan. 2012. Optimal crowdsourcing contests. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, page 856–868, USA. Society for Industrial and Applied Mathematics.

Yingying Chen, Jacob Long, Jungmi Jun, Sei-Hill Kim, Ali Zain, and Colin Piacentine. 2023. Anti-intellectualism amid the COVID-19 pandemic: The discursive elements and sources of anti-Fauci tweets. *Public Understanding of Science*, page 096366252211462.

Ruijia Cheng, Aayushi Dangol, Frances Marie Tabio Ello, Lingyu Wang, and Sayamindu Dasgupta. 2023a. Concepts, practices, and perspectives for developing computational data literacy: Insights from workshops with a new data programming system. In *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*, IDC '23, page 100–111, New York, NY, USA. Association for Computing Machinery.

Ruijia Cheng, Sayamindu Dasgupta, and Benjamin Mako Hill. 2022a. How interest-driven content creation constrains opportunities for informal learning: A case study on novices' use of data structures in scratch. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems - CHI '22*, New Orleans, LA, USA. ACM Press.

Ruijia Cheng, Sayamindu Dasgupta, and Benjamin Mako Hill. 2022b. How Interest-Driven Content Creation Shapes Opportunities for Informal Learning in Scratch: A Case Study on Novices' Use of Data Structures. In *CHI Conference on Human Factors in Computing Systems*, pages 1–16, New Orleans LA USA. ACM.

Ruijia Cheng and Benjamin Mako Hill. 2022. Many destinations, many pathways: A quantitative analysis of legitimate peripheral participation in scratch. *Proc. ACM Hum.-Comput. Interact.*, 6(CSCW2).

Ruijia Cheng, Ruotong Wang, Thomas Zimmermann, and Denae Ford. 2023b. "it would work for me too": How online communities shape software developers' trust in ai-powered code generation tools.

Ruijia Cheng and Mark Zachry. 2020a. Building Community Knowledge In Online Competitions: Motivation, Practices and Challenges. In *Proc. ACM Hum.-Comput. Interact.4, CSCW2*, volume 4.

Ruijia Cheng and Mark Zachry. 2020b. Building Community Knowledge In Online Competitions: Motivation, Practices and Challenges. In *Proc. ACM Hum.-Comput. Interact.4, CSCW2*, volume 4.

Ruijia Cheng, Ziwen Zeng, Maysnow Liu, and Steven Dow. 2020. Critique me: Exploring how creators publicly request feedback in an online critique community. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW2).

Henry William Chesbrough. 2003. *Open innovation: The new imperative for creating and profiting from technology*. Harvard Business Press.

Matteo Cinelli, Walter Quattrociocchi, Alessandro Galeazzi, Carlo Michele Valensise, Emanuele Brugnoli, Ana Lucia Schmidt, Paola Zola, Fabiana Zollo, and Antonio Scala. 2020. The COVID-19 social media infodemic. *Scientific Reports*, 10(1):16598.

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

Allan Collins, John Seely Brown, and Ann Holum. 1991. Cognitive apprenticeship: Making thinking visible. *American educator*, 15(3):6–11.

James Collins. 1995. Literacy and literacies. *Annual Review of Anthropology*, 24(1):75–93.

Anamaria Crisan, Brittany Fiore-Gartland, and Melanie K. Tory. 2020. Passing the data baton : A retrospective analysis on data science work and workers. *IEEE Transactions on Visualization and Computer Graphics*, 27:1860–1870.

CrowdFlower. 2017. CrowdFlower. Data Scientist Report.

Kevin Crowston and Isabelle Fagnot. 2018. Stages of motivation for contributing user-generated content: A theory and empirical test. *International Journal of Human-Computer Studies*, 109:89–101.

Diego F. Cuadros, F. DeWolfe Miller, Susanne Awad, Philip Coule, and Neil J. MacKinnon. 2022. Analysis of Vaccination Rates and New COVID-19 Infections by US County, July-August 2021. *JAMA Network Open*, 5(2):e2147915.

Limeng Cui and Dongwon Lee. 2020. CoAID: COVID-19 Healthcare Misinformation Dataset. ArXiv:2006.00885 [cs].

Hai Min Dai, Timothy Teo, and Natasha Anne Rappa. 2022. The role of gender and employment status in mooc learning: An exploratory study. *Journal of Computer Assisted Learning*, 38(5):1360–1370.

Sayamindu Dasgupta. 2013. From Surveys to Collaborative Art: Enabling Children to Program with Online Data. In *Proceedings of the 12th International Conference on Interaction Design and Children (IDC '13)*, pages 28–35, New York, NY. ACM.

Sayamindu Dasgupta. 2016. *Children as Data Scientists: Explorations in Creating, Thinking, and Learning*. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Sayamindu Dasgupta, William Hale, Andrés Monroy-Hernández, and Benjamin Mako Hill. 2016a. Remixing as a pathway to computational thinking. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, CSCW '16, page 1438–1449, New York, NY, USA. Association for Computing Machinery.

Sayamindu Dasgupta, William Hale, Andrés Monroy-Hernández, and Benjamin Mako Hill. 2016b. Remixing as a Pathway to Computational Thinking. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*, pages 1438–1449, New York, NY. ACM.

Sayamindu Dasgupta and Benjamin Mako Hill. 2017a. Learning to Code in Localized Programming Languages. In *Proceedings of the Fourth ACM Conference on Learning @ Scale (L@S '17)*, pages 33–39, New York, NY. ACM.

Sayamindu Dasgupta and Benjamin Mako Hill. 2017b. Scratch Community Blocks: Supporting Children as Data Scientists. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*, pages 3620–3631, New York, New York. ACM Press.

Sayamindu Dasgupta and Benjamin Mako Hill. 2018a. How "Wide Walls" Can Increase Engagement: Evidence from a Natural Experiment in Scratch. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*, pages 361:1–361:11, New York, New York. ACM.

Sayamindu Dasgupta and Benjamin Mako Hill. 2018b. How "wide walls" can increase engagement: evidence from a natural experiment in scratch. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–11.

Oceans of Data Institute staff. 2016. Building Global Interest in Data Literacy: A Dialogue. Technical report, Oceans of Data Institute, Education Development Center, Inc.

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

Peter J. Denning and Matti Tedre. 2019. *Computational Thinking*. MIT Press, Cambridge, MA.

Sebastian Deterding, Rilla Khaled, Lenard E Nacke, and Dan Dixon. 2011. Gamification: Toward a definition in chi 2011 gamification workshop proceedings, vancouver, bc, canada. *Online: http://gamification-research. org/wp-content/uploads/2011/04/02-Deterding-Khaled-Nacke-Dixon. pdf [14.4. 2017]*.

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

Catherine D'Ignazio and Rahul Bhargava. 2015. Approaches to Building Big Data Literacy. In *Proceedings of the Bloomberg Data for Good Exchange Conference 2015*, New York, N.Y.

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

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

Andrea A. diSessa. 1986. Models of Computation. In Donald A. Norman and Stephen W. Draper, editors, *User-Centered System Design: New Perspectives in Human-Computer Interaction*, pages 201–218. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Paul Dourish and Edgar Gómez Cruz. 2018. Datafication and data fiction: Narrating data and narrating with data. *Big Data & Society*, 5(2):2053951718784083.

Juliana Elisa Raffaghelli. 2020. Is Data Literacy a Catalyst of Social Justice? A Response from Nine Data Literacy Initiatives in Higher Education. *Education Sciences*, 10(9):233.

Sarah Evans, Katie Davis, Abigail Evans, Julie Ann Campbell, David P Randall, Kodlee Yin, and Cecilia Aragon. 2017. More than peer production: Fanfiction communities as sites of distributed mentoring. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 259–272.

Melanie Feinberg. 2017. A Design Perspective on Data. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2952–2963, Denver, Colorado, USA. Association for Computing Machinery.

Deborah A. Fields, Michael Giang, and Yasmin Kafai. 2014. Programming in the Wild: Trends in Youth Computational Participation in the Online Scratch Community. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, WiPSCE '14, pages 2–11, New York, NY, USA. ACM.

Deborah A. Fields, Katarina Pantic, and Yasmin B. Kafai. 2015. "I Have a Tutorial for This": The Language of Online Peer Support in the Scratch Programming Community. In *Proceedings of the 14th International Conference on Interaction Design and Children*, IDC '15, pages 229–238, New York, NY, USA. ACM.

Casey Fiesler, Shannon Morrison, R. Benjamin Shapiro, and Amy S. Bruckman. 2017a. Growing their own: Legitimate peripheral participation for computational learning in an online fandom community. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, page 1375–1386, New York, NY, USA. Association for Computing Machinery.

Casey Fiesler, Shannon Morrison, R. Benjamin Shapiro, and Amy S. Bruckman. 2017b. Growing Their Own: Legitimate Peripheral Participation for Computational Learning in an Online Fandom Community. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 1375–1386, Portland Oregon USA. ACM.

Casey Fiesler, Shannon Morrison, R. Benjamin Shapiro, and Amy S. Bruckman. 2017c. Growing their own: Legitimate peripheral participation for computational learning in an online fandom community. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, page 1375–1386, New York, NY, USA. Association for Computing Machinery.

Alexandre de Figueiredo, Clarissa Simas, Emilie Karafillakis, Pauline Paterson, and Heidi J Larson. 2020. Mapping global trends in vaccine confidence and investigating barriers to vaccine uptake: a large-scale retrospective temporal modelling study. *The Lancet*, 396(10255):898–908.

William Finzer and Dan Damelin. 2015. Building the CODAP Community. *@Concord*, 19(2):8–9.

Eureka Foong, Steven P. Dow, Brian P. Bailey, and Elizabeth M. Gerber. 2017. Online feedback exchange: A framework for understanding the socio-psychological factors. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 4454–4467, New York, NY, USA. Association for Computing Machinery.

Denae Ford, Kristina Lustig, Jeremy Banks, and Chris Parnin. 2018a. "We Don't Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–12, Montreal QC, Canada. ACM Press.

Denae Ford, Kristina Lustig, Jeremy Banks, and Chris Parnin. 2018b. "We Don't Do That Here": How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–12, Montreal QC, Canada. ACM Press.

Denae Ford, Justin Smith, Philip J Guo, and Chris Parnin. 2016. Paradise unplugged: Identifying barriers for female participation on stack overflow. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 846–857.

N. Fraser. 2015. Ten things we've learned from Blockly. In *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, pages 49–50.

Johann Füller, Katja Hutter, and Rita Faullant. 2011. Why co-creation experience matters? creative experience and its impact on the quantity and quality of creative contributions. *R&D Management*, 41(3):259–273.

Alexandra Funke and Katharina Geldreich. 2017. Gender Differences in Scratch Programs of Primary School Children. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, pages 57–64, Nijmegen Netherlands. ACM.

Emilia F. Gan, Benjamin Mako Hill, and Sayamindu Dasgupta. 2018. Gender, Feedback, and Learners' Decisions to Share Their Creative Computing Projects. *Proceedings of the ACM: Human-Computer Interaction*, 2(CSCW):54:1–54:23.

James Paul Gee. 2005. Semiotic social spaces and affinity spaces. *Beyond communities of practice language power and social context*, 214232.

James Paul Gee. 2006. *Situated Language and Learning: A Critique of Traditional Schooling*, reprinted edition. Literacies. Routledge, New York.

Sarah Gilbert. 2016. Learning in a twitter-based community of practice: an exploration of knowledge exchange as a motivation for participation in# hcsmca. *Information, Communication & Society*, 19(9):1214–1232.

Ashish Goel and Latika Gupta. 2020. Social Media in the Times of COVID-19. *JCR: Journal of Clinical Rheumatology*, 26(6):220–223.

Bette Gray. 2004. Informal learning in an online community of practice. *Journal of Distance Education*, 19(1):20–35.

Colin M. Gray and Yubo Kou. 2019. Co-Producing, Curating, and Defining Design Knowledge in an Online Practitioner Community. *CoDesign*, 15(1):41–58.

Mary L. Gray, Siddharth Suri, Syed Shoaib Ali, and Deepti Kulkarni. 2016. The crowd is a collaborative network. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, CSCW '16, page 134–147, New York, NY, USA. Association for Computing Machinery.

Christine Greenhow and Beth Robelia. 2009. Informal learning and identity formation in online social networks. *Learning, Media and Technology*, 34(2):119–140.

Anatoliy Gruzd, Barry Wellman, and Yuri Takhteyev. 2011. Imagining twitter as an imagined community. *American Behavioral Scientist*, 55(10):1294–1318.

Greg Guest, Kathleen M MacQueen, and Emily E Namey. 2011. *Applied thematic analysis*. Sage Publications.

Keith Gunaratne, Eric A. Coomes, and Hourmazd Haghbayan. 2019. Temporal trends in anti-vaccine discourse on Twitter. *Vaccine*, 37(35):4867–4871.

Aaron Halfaker, R Stuart Geiger, Jonathan T Morgan, and John Riedl. 2013a. The rise and decline of an open collaboration system: How wikipedia's reaction to popularity is causing its decline. *American Behavioral Scientist*, 57(5):664–688.

Aaron Halfaker, Os Keyes, and Dario Taraborelli. 2013b. Making peripheral participation legitimate: reader engagement experiments in wikipedia. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 849–860.

Jaron Harambam and Stef Aupers. 2021. From the unbelievable to the undeniable: Epistemological pluralism, or how conspiracy theorists legitimate their extraordinary truth claims. *European Journal of Cultural Studies*, 24(4):990–1008.

Samantha Hautea, Sayamindu Dasgupta, and Benjamin Mako Hill. 2017a. Youth Perspectives on Critical Data Literacies. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 919–930, Denver Colorado USA. ACM.

Samantha Hautea, Sayamindu Dasgupta, and Benjamin Mako Hill. 2017b. Youth perspectives on critical data literacies. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 919–930.

K. Hayawi, S. Shahriar, M. A. Serhani, I. Taleb, and S. S. Mathew. 2022. ANTi-Vax: a novel Twitter dataset for COVID-19 vaccine misinformation detection. *Public Health*, 203:23–30.

Felienne Hermans, Alaaeddin Swidan, Efthimia Aivaloglou, and Marileen Smit. 2018. Thinking out of the Box: Comparing Metaphors for Variables in Programming Education. In *Proceedings of the 13th Workshop in Primary and Secondary Computing Education on - WiPSCE '18*, pages 1–8, Potsdam, Germany. ACM Press.

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

Benjamin Mako Hill and Andrés Monroy-Hernández. 2013. The cost of collaboration for code and art: Evidence from a remixing community. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1035–1046.

Benjamin Mako Hill and Andrés Monroy-Hernández. 2017. A longitudinal dataset of five years of public activity in the Scratch online community. *Scientific Data*, 4:170002.

Benjamin Mako Hill and Andrés Monroy-Hernández. 2013. The Remixing Dilemma The Trade-Off Between Generativity and Originality. *American Behavioral Scientist*, 57(5):643–663.

Eric von Hippel and Georg von Krogh. 2003. Exploring the open source software phenomenon: issues for organization science.

Beth L. Hoffman, Jason B. Colditz, Ariel Shensa, Riley Wolynn, Sanya Bathla Taneja, Elizabeth M. Felter, Todd Wolynn, and Jaime E. Sidani. 2021. #DoctorsSpeakUp: Lessons learned from a pro-vaccine Twitter event. *Vaccine*, 39(19):2684–2691.

Beth L. Hoffman, Elizabeth M. Felter, Kar-Hai Chu, Ariel Shensa, Chad Hermann, Todd Wolynn, Daria Williams, and Brian A. Primack. 2019. It's not all about autism: The emerging landscape of anti-vaccination sentiment on Facebook. *Vaccine*, 37(16):2216–2223.

Maya Holikatti, Shagun Jhaver, and Neha Kumar. 2019. Learning to airbnb by engaging in online communities of practice. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–19.

Allison Marie Horst, Alison Presmanes Hill, and Kristen B. Gorman. 2020. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*.

John Joseph Horton and Lydia B. Chilton. 2010. The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, EC '10, page 209–218, New York, NY, USA. Association for Computing Machinery.

Peter Hotez, Carolina Batista, Onder Ergonul, J Peter Figueroa, Sarah Gilbert, Mayda Gursel, Mazen Hassanain, Gagandeep Kang, Jerome H Kim, Bhavna Lall, Heidi Larson, Denise Naniche, Timothy Sheahan, Shmuel Shoham, Annelies Wilder-Smith, Nathalie Strub-Wourgaft, Prashant Yadav, and Maria Elena Bottazzi. 2021. Correcting COVID-19 vaccine misinformation. *EClinicalMedicine*, 33:100780.

Hui-mei Justina Hsu. 2014. Gender Differences in Scratch Game Design. In *Proceedings of the 2014*

International Conference on Information, Business and Education Technology, Beijing, China. Atlantis Press.

Shih-Wen Huang, Minhyang Suh, Benjamin Mako Hill, and Gary Hsieh. 2015. How activists are both born and made: An analysis of users on change. org. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 211–220.

Bernardo A Huberman, Daniel M Romero, and Fang Wu. 2009. Crowdsourcing, attention and productivity. *Journal of Information Science*, 35(6):758–765.

Julie S. Hui, Matthew W. Easterday, and Elizabeth M. Gerber. 2019. Distributed Apprenticeship in Online Communities. *Human– Computer Interaction*, 34(4):328–378.

Julie S. Hui, Darren Gergle, and Elizabeth M. Gerber. 2018a. IntroAssist: A Tool to Support Writing Introductory Help Requests. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–13, Montreal QC, Canada. ACM Press.

Julie S Hui, Darren Gergle, and Elizabeth M Gerber. 2018b. Introassist: A tool to support writing introductory help requests. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Jessica Hullman, Xiaoli Qiao, Michael Correll, Alex Kale, and Matthew Kay. 2019. In Pursuit of Error: A Survey of Uncertainty Visualization Evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):903–913.

Katja Hutter, Julia Hautz, Johann Füller, Julia Mueller, and Kurt Matzler. 2011. Communitition: The tension between competition and collaboration in community-based design contests. *Creativity and innovation management*, 20(1):3–21.

Mizuko Ito, editor. 2009. *Hanging Out, Messing Around, and Geeking Out: Kids Living and Learning with New Media*. The MIT Press.

Mizuko Ito. 2013. *Hanging out, messing around, and geeking out: Kids living and learning with new media*. The MIT Press.

Mizuko Ito, Kris Gutiérrez, Sonia Livingstone, Bill Penuel, Jean Rhodes, Katie Salen, Juliet Schor, Julian Sefton-Green, and S. Craig Watkins. 2013. *Connected Learning*. BookBaby, Cork.

Mizuko Ito, Crystle Martin, Rachel Cody Pfister, Matthew H Rafalow, Katie Salen, and Amanda Wortman. 2018. *Affinity online: How connection and shared interest fuel learning*, volume 2. NYU Press.

Amelia M. Jamison, David A. Broniatowski, Mark Dredze, Zach Wood-Doughty, DureAden Khan, and Sandra Crouse Quinn. 2020. Vaccine-related advertising in the Facebook Ad Archive. *Vaccine*, 38(3):512–520.

Henry Jenkins, Mizuko Itō, and danah boyd. 2016. *Participatory Culture in a Networked Era: A Conversation on Youth, Learning, Commerce, and Politics*.

Henry Jenkins, Ravi Purushotma, Margaret Weigel, Katie Clinton, and Alice J. Robison. 2009. *Confronting the Challenges of Participatory Culture: Media Education for the 21st Century*. The John D. and Catherine T. MacArthur Foundation Reports on Digital Media and Learning. The MIT Press, Cambridge, MA.

Joseph M Juran. 1954. Universals in Management Planning and Controlling. *Management Review*, 43(11):748–761.

Yasmin Kafai. 2017. Connected Gaming: An Inclusive Perspective for Serious Gaming. *International Journal of Serious Games*, 4(3).

Yasmin B. Kafai. 2016. From computational thinking to computational participation in K–12 education. *Communications of the ACM*, 59(8):26–27.

Yasmin B. Kafai and Quinn Burke. 2014. *Connected Code: Why Children Need to Learn Programming*. MIT Press, Cambridge, Massachusetts.

Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer. 2012. Enterprise Data Analysis and Visualization: An Interview Study. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2917–2926.

Tobias Kauer, Marian Dörk, Arran L. Ridley, and Benjamin Bach. 2021. The public life of data: Investigating reactions to visualizations on reddit. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA. Association for Computing Machinery.

Allison Kennedy, Katherine LaVail, Glen Nowak, Michelle Basket, and Sarah Landry. 2011. Confidence About Vaccines In The United States: Understanding Parents' Perceptions. *Health Affairs*, 30(6):1151–1159.

Joy Kim, Maneesh Agrawala, and Michael S. Bernstein. 2017a. Mosaic: Designing Online Creative Communities for Sharing Works-in-Progress. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, pages 246–258, New York, NY, USA. ACM.

Joy Kim, Maneesh Agrawala, and Michael S. Bernstein. 2017b. Mosaic: Designing online creative communities for sharing works-in-progress. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, page 246–258, New York, NY, USA. Association for Computing Machinery.

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

Yasmine Kotturi and McKayla Kingston. 2019. Why Do Designers in the "Wild" Wait to Seek Feedback until Later in Their Design Process? In *Proceedings of the 2019 on Creativity and Cognition*, pages 541–546, San Diego CA USA. ACM.

Yubo Kou and Colin M. Gray. 2017a. Supporting Distributed Critique through Interpretation and Sense-Making in an Online Creative Community. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):1–18.

Yubo Kou and Colin M Gray. 2017b. Supporting distributed critique through interpretation and sense-making in an online creative community. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):60.

Yubo Kou and Colin M. Gray. 2018. "What Do You Recommend a Complete Beginner like Me to Practice?": Professional Self-Disclosure in an Online Community. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–24.

Yubo Kou, Colin M Gray, Austin L Toombs, and Robin S Adams. 2018. Understanding social roles in an online community of volatile practice: A study of user experience practitioners on reddit. *ACM Transactions on Social Computing*, 1(4):1–22.

Robert E Kraut and Paul Resnick. 2012. *Building successful online communities: Evidence-based social design*. Mit Press.

Robert E. Kraut, Paul Resnick, and Sara Kiesler. 2011. *Building Successful Online Communities: Evidence-Based Social Design*. MIT Press, Cambridge, Mass.

Georg von Krogha, S. Spaeth, and K. Lakhani. 2003. Community, joining, and specialization in open source software innovation: a case study. *Social Science Research Network*.

Sean Kross and Philip J. Guo. 2019. Practitioners Teaching Data Science in Industry and Academia: Expectations, Workflows, and Challenges. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, Glasgow Scotland Uk. ACM.

Chinmay Kulkarni, Steven P Dow, and Scott R Klemmer. 2014. Early and Repeated Exposure to Examples Improves Creative Work. In *Design Thinking Research*, pages 49–62. Springer, Cham.

Navin Kumar, Isabel Corpus, Meher Hans, Nikhil Harle, Nan Yang, Curtis McDonald, Shinpei Nakamura Sakai, Kamila Janmohamed, Keyu Chen, Frederick L. Altice, Weiming Tang, Jason L. Schwartz, S. Mo Jones-Jang, Koustuv Saha, Shahan Ali Memon, Chris T. Bauch, Munmun De Choudhury, Orestis Papakyriakopoulos, Joseph D. Tucker, Abhay Goyal, Aman Tyagi, Kaveh Khoshnood, and Saad Omer. 2022. COVID-19 vaccine perceptions in the initial phases of US vaccine roll-out: an observational study on reddit. *BMC Public Health*, 22(1):446.

Karim R Lakhani, David A Garvin, and Eric Lonstein. 2010. Topcoder (a): Developing software through crowdsourcing. *Harvard Business School General Management Unit Case*, (610-032).

Thomas D. LaToza, Micky Chen, Luxi Jiang, Mengyao Zhao, and André van der Hoek. 2015. Borrowing from the crowd: A study of recombination in software design competitions. In *Proceedings of the 37th International Conference on Software Engineering - Volume 1*, ICSE '15, page 551–562. IEEE Press.

Jean Lave and Etienne Wenger. 1991. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, Cambridge, UK.

Jean Lave, Etienne Wenger, et al. 1991. *Situated learning: Legitimate peripheral participation*. Cambridge university press.

Crystal Lee, Tanya Yang, Gabrielle D Inchoco, Graham M. Jones, and Arvind Satyanarayan. 2021. Viral Visualizations: How Coronavirus Skeptics Use Orthodox Data Practices to Promote Unorthodox Science Online. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–18, Yokohama Japan. ACM.

Victor Lee and Michelle Wilkerson. 2018. Data Use by Middle and Secondary Students in the Digital Age: A Status Report and Future Prospects.

Victor R. Lee. 2013. The Quantified Self (QS) Movement and Some Emerging Opportunities for the Educational Technology Field. *Educational Technology*, 53(6):39–42.

Victor R. Lee, Daniel R. Pimentel, Rahul Bhargava, and Catherine D'Ignazio. 2022. Taking data feminism to school: A synthesis and review of pre-collegiate data science education projects. *British Journal of Educational Technology*, 53(5):1096–1113.

Jan Marco Leimeister, Helmut Krcmar, Ulrich Bretschneider, and Winfried Ebner. 2008. Leveraging the wisdom of crowds: Designing an it-supported ideas competition for an erp software company.

Carson K. Leung, Yubo Chen, Calvin S.H. Hoi, Siyuan Shang, Yan Wen, and Alfredo Cuzzocrea. 2020. Big Data Visualization and Visual Analytics of COVID-19 Data. In *2020 24th International Conference Information Visualisation (IV)*, pages 415–420, Melbourne, Australia. IEEE.

Guo Li, Haiyi Zhu, Tun Lu, Xianghua Ding, and Ning Gu. 2015a. Is It Good to Be Like Wikipedia?: Exploring the Trade-Offs of Introducing Collaborative Editing Model to Q&A Sites. In *Proceedings of*

the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '15, pages 1080–1091, Vancouver, BC, Canada. ACM Press.

Guo Li, Haiyi Zhu, Tun Lu, Xianghua Ding, and Ning Gu. 2015b. Is it good to be like wikipedia? exploring the trade-offs of introducing collaborative editing model to q&a sites. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, page 1080–1091, New York, NY, USA. Association for Computing Machinery.

Bridget Lockyer, Shahid Islam, Aamnah Rahman, Josie Dickerson, Kate Pickett, Trevor Sheldon, John Wright, Rosemary McEachan, Laura Sheard, and the Bradford Institute for Health Research Covid-19 Scientific Advisory Group. 2021. Understanding COVID-19 misinformation and vaccine hesitancy in context: Findings from a qualitative study involving citizens in Bradford, UK. *Health Expectations*, 24(4):1158–1167.

Sahil Loomba, Alexandre de Figueiredo, Simon J. Piatek, Kristen de Graaf, and Heidi J. Larson. 2021. Measuring the impact of COVID-19 vaccine misinformation on vaccination intent in the UK and USA. *Nature Human Behaviour*, 5(3):337–348.

Kai Lu, Wenjun Zhou, and Xuehua Wang. 2014. Social network of the competing crowd. In *2014 International Conference on Behavioral, Economic, and Socio-Cultural Computing (BESC2014)*, pages 1–7. IEEE.

Kurt Luther, Kevin Ziegler, Kelly E Caine, and Amy Bruckman. 2009. Predicting successful completion of online collaborative animation projects. In *Proceedings of the seventh ACM conference on Creativity and cognition*, pages 391–392.

Joanne Chen Lyu, Eileen Le Han, and Garving K Luli. 2021. COVID-19 Vaccine–Related Discussion on Twitter: Topic Modeling and Sentiment Analysis. *Journal of Medical Internet Research*, 23(6):e24435.

Noni E. MacDonald. 2015. Vaccine hesitancy: Definition, scope and determinants. *Vaccine*, 33(34):4161–4164.

Leticia Santos Machado, Ricardo R. M. Melo, and Cleidson R. B. de Souza. 2019. The role of platform moderators in software crowdsourcing projects. In *Proceedings of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE '19, page 119–122. IEEE Press.

Stephen MacNeil, Zijian Ding, Kexin Quan, Thomas j Parashos, Yajie Sun, and Steven P Dow. 2021. Framing creative work: Helping novices frame better problems through interactive scaffolding. In *Creativity and Cognition*, pages 1–10.

John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. The Scratch Programming Language and Environment. *Trans. Comput. Educ.*, 10(4):16:1–16:15.

Ekaterina Malova. 2021. Understanding online conversations about COVID-19 vaccine on Twitter: vaccine hesitancy amid the public health crisis. *Communication Research Reports*, 38(5):346–356.

Annette Markham. 2012. Fabrication as Ethical Practice. *Information, Communication & Society*, 15(3):334–353.

Jennifer Marlow and Laura Dabbish. 2014a. From rookie to all-star: Professional development in a graphic design social networking site. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '14*, pages 922–933, Baltimore, Maryland, USA. ACM Press.

Jennifer Marlow and Laura Dabbish. 2014b. From Rookie to All-Star: Professional Development in a Graphic Design Social Networking Site. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '14*, pages 922–933, Baltimore, Maryland, USA. ACM Press.

Christopher A. Martin, Colette Marshall, Prashanth Patel, Charles Goss, David R. Jenkins, Claire Ellwood, Linda Barton, Arthur Price, Nigel J. Brunskill, Kamlesh Khunti, and Manish Pareek. 2021. SARS-CoV-2 vaccine uptake in a multi-ethnic UK healthcare workforce: A cross-sectional study. *PLOS Medicine*, 18(11):e1003823.

Christina Masden and W. Keith Edwards. 2015. Understanding the role of community in online dating. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 535–544, New York, NY, USA. Association for Computing Machinery.

J. Nathan Matias, Sayamindu Dasgupta, and Benjamin Mako Hill. 2016. Skill Progression in Scratch Revisited. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*, pages 1486–1490, New York, NY. ACM Press.

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

Brian McInnis, Xiaotong Tone Xu, and Steven P Dow. 2018. How features of a civic design competition influences the collective understanding of a problem. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):120.

David W McMillan and David M Chavis. 1986. Sense of community: A definition and theory. *Journal of community psychology*, 14(1):6–23.

Katrina Megget. 2020. Even covid-19 can't kill the anti-vaccination movement. *BMJ*, page m2184.

R. K. Merton. 1968. The Matthew Effect in Science: The Reward and Communication Systems of Science Are Considered. *Science*, 159(3810):56–63.

Elena Milani, Emma Weitkamp, and Peter Webb. 2020. The Visual Vaccine Debate on Twitter: A Social Network Analysis. *Media and Communication*, 8(2):364–375.

Benny Moldovanu and Aner Sela. 2001. The optimal allocation of prizes in contests. *American Economic Review*, 91(3):542–558.

Andrés Monroy-Hernández. 2007. ScratchR: Sharing user-generated programmable media. In *Proceedings of the 6th International Conference on Interaction Design and Children*, IDC '07, pages 167–168, New York, NY. ACM.

John Morgan and Richard Wang. 2010. Tournaments for ideas. *California management review*, 52(2):77–97.

Gabriel Mugar, Carsten Østerlund, Katie DeVries Hassman, Kevin Crowston, and Corey Brian Jackson. 2014. Planet hunters and seafloor explorers: Legitimate peripheral participation through practice proxies in online citizen science. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '14, page 109–119, New York, NY, USA. Association for Computing Machinery.

Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How data science workers work with data: Discovery, capture, curation, design, creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Goran Muric, Yusong Wu, and Emilio Ferrara. 2021. COVID-19 Vaccine Hesitancy on Social Media: Building a Public Twitter Dataset of Anti-vaccine Content, Vaccine Misinformation and Conspiracies. Publisher: arXiv Version Number: 2.

S. Nag, I. Heffan, A. Saenz-Otero, and M. Lydon. 2012. Spheres zero robotics software development: Lessons on crowdsourcing and collaborative competition. In *2012 IEEE Aerospace Conference*, pages 1–17.

Jakob Nielsen. 2006. Participation inequality: Encouraging more users to contribute. *http://www. useit. com/alertbox/participation_inequality. html*.

Ryo Okubo, Takashi Yoshioka, Satoko Ohfuji, Takahiro Matsuo, and Takahiro Tabuchi. 2021. COVID-19 Vaccine Hesitancy and Its Associated Factors in Japan. *Vaccines*, 9(6):662.

Katherine Panciera, Aaron Halfaker, and Loren Terveen. 2009. Wikipedians are born, not made: a study of power editors on wikipedia. In *Proceedings of the ACM 2009 international conference on Supporting group work*, pages 51–60.

Seymour Papert. 1980a. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York, NY.

Seymour Papert. 1991a. Situating Constructionism. In Idit Harel and Seymour Papert, editors, *Constructionism*, volume 36, pages 1–11. Ablex Publishing, New York, NY, US.

Seymour Papert. 1991b. Situating constructionism. In Idit Harel and Seymour Papert, editors, *Constructionism*, volume 36, pages 1–11. Ablex Publishing, New York, NY, US.

Seymour Papert. 1993. *Mindstorms: Children, Computers, and Powerful Ideas*, 2nd ed edition. Basic Books, New York.

Seymour A Papert. 1980b. *Mindstorms: Children, computers, and powerful ideas*. Basic books.

Miranda C. Parker and Leigh Ann DeLyser. 2017. Concepts and Practices: Designing and Developing A Modern K-12 CS Framework. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '17, pages 453–458, New York, NY, USA. Association for Computing Machinery.

Daniel Pipes. 1997. *Conspiracy: how the paranoid style flourishes and where it comes from*. Free Press, New York.

Javier Calzada Prado and Miguel Ángel Marzal. 2013. Incorporating data literacy into information literacy programs: Core competencies and contents. *Libri*, 63(2):123–134.

Jennifer Preece and Ben Shneiderman. 2009. The reader-to-leader framework: Motivating technology-mediated social participation. *AIS transactions on human-computer interaction*, 1(1):13–32.

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

Mitchel Resnick. 2016. Designing for Wide Walls. Publication Title: Design.blog.

Mitchel Resnick, Stephen Benton, Moose Crossing, and Amy Bruckman. 1998. Moose crossing: Construction, community, and learning in a networked virtual world for kids.

Mitchel Resnick, Amy Bruckman, and Fred Martin. 1996a. Pianos not stereos: Creating computational construction kits. *interactions*, 3(5):40–50.

Mitchel Resnick, Amy Bruckman, and Fred Martin. 1996b. Pianos not stereos: Creating computational construction kits. *interactions*, 3(5):40–50.

Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009a. Scratch: Programming for all. *Communications of the ACM*, 52(11):60–67.

Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009b. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67.

Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009c. Scratch: Programming for All. *Communications of the ACM*, 52(11):60–67.

Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. 2005. Design principles for tools to support creative thinking.

Mitchel Resnick and Brian Silverman. 2005. Some Reflections on Designing Construction Kits for Kids. In *Proceedings of the 2005 Conference on Interaction Design and Children (IDC '05)*, pages 117–122, New York, NY. ACM.

Gabriela T. Richard and Yasmin B. Kafai. 2016. Blind spots in youth diy programming: Examining diversity in creators, content, and comments within the scratch online community. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 1473–1485, New York, NY, USA. Association for Computing Machinery.

Ganit Richter, Daphne R Raban, and Sheizaf Rafaeli. 2015. Studying gamification: the effect of rewards and incentives on motivation. In *Gamification in education and business*, pages 21–46. Springer.

Ricarose Roque, Yasmin Kafai, and Deborah Fields. 2012. From tools to communities: Designs to support online creative collaboration in scratch. In *Proceedings of the 11th International Conference on Interaction Design and Children*, IDC '12, page 220–223, New York, NY, USA. Association for Computing Machinery.

Ricarose Roque, Natalie Rusk, and Amos Blanton. 2013. Youth Roles and Leadership in an Online Creative Community. In *Proceedings of the 10th International Conference on Computer-Supported Collaborative Learning*, volume 1, pages 399–405, Madison, WI. International Society of the Learning Sciences (ISLS).

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

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

Adam Rule, Ian Drosos, Aurélien Tabard, and James D. Hollan. 2018a. Aiding collaborative reuse of computational notebooks with annotated cell folding. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW).

Adam Rule, Aurélien Tabard, and James D. Hollan. 2018b. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA. Association for Computing Machinery.

Jean Salac and Diana Franklin. 2020. If They Build It, Will They Understand It? Exploring the Relationship between Student Code and Performance. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, pages 473–479, Trondheim Norway. ACM.

Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*.

Christopher Scaffidi and Christopher Chambers. 2012. Skill progression demonstrated by users in the scratch animation environment. *International Journal of Human-Computer Interaction*, 28(6):383–398.

Christopher Scaffidi, Aniket Dahotre, and Yan Zhang. 2012. How Well Do Online Forums Facilitate Discussion and Collaboration among Novice Animation Programmers? In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education - SIGCSE '12*, page 191, Raleigh, North Carolina, USA. ACM Press.

Marlene Scardamalia. 2002. Collective cognitive responsibility for the advancement of knowledge. *Liberal education in a knowledge society*, 97:67–98.

Eva Schernhammer, Jakob Weitzer, Manfred D Laubichler, Brenda M Birmann, Martin Bertau, Lukas Zenk, Guido Caniglia, Carlo C Jäger, and Gerald Steiner. 2022. Correlates of COVID-19 vaccine hesitancy in Austria: trust and the government. *Journal of Public Health*, 44(1):e106–e116.

Samantha Shorey, Benjamin Mako Hill, and Samuel Woolley. 2020a. From Hanging out to Figuring It out: Socializing Online as a Pathway to Computational Thinking. *New Media & Society*, page 1461444820923674.

Samantha Shorey, Benjamin Mako Hill, and Samuel Woolley. 2020b. From hanging out to figuring it out: Socializing online as a pathway to computational thinking. *New Media & Society*, page 1461444820923674.

Nischal Shrestha, Titus Barik, and Chris Parnin. 2021. Remote, but connected: How# tidytuesday provides an online community of practice for data scientists. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–31.

Judith D. Singer and John B. Willett. 2003. *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. Oxford University Press, Oxford ; New York.

Cleidson R. B. de Souza, Leticia S. Machado, and Ricardo Rodrigo M. Melo. 2020. On moderating software crowdsourcing challenges. *Proc. ACM Hum.-Comput. Interact.*, 4(GROUP).

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

Rabindranath Tagore. 1930. *Sahaj Path - Part 1*. Visva-Bharati Granthanbibhag, Santiniketan, India.

Yla Tausczik and Ping Wang. 2017a. To Share, or Not to Share?: Community-Level Collaboration in Open Innovation Contests. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):1–23.

Yla Tausczik and Ping Wang. 2017b. To share, or not to share?: Community-level collaboration in open innovation contests. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):100.

Yla R. Tausczik, Aniket Kittur, and Robert E. Kraut. 2014a. Collaborative Problem Solving: A Study of MathOverflow. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '14*, pages 355–367, Baltimore, Maryland, USA. ACM Press.

Yla R. Tausczik, Aniket Kittur, and Robert E. Kraut. 2014b. Collaborative problem solving: A study of mathoverflow. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '14, page 355–367, New York, NY, USA. Association for Computing Machinery.

Yla R. Tausczik and James W. Pennebaker. 2010. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, 29(1):24–54.

Giovanni Maria Troiano, Qinyu Chen, Ángela Vargas Alba, Gregorio Robles, Gillian Smith, Michael Cassidy, Eli Tucker-Raymond, Gillian Puttick, and Casper Harteveld. 2020. Exploring How Game Genre in Student-Designed Games Influences Computational Thinking Development. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–17, Honolulu HI USA. ACM.

Jan E. Trost. 1986. Statistically Nonrepresentative Stratified Sampling: A Sampling Technique for Qualitative Studies. *Qualitative Sociology*, 9(1):54–57.

Jonathan Y. Tsou. 2006. Genetic epistemology and piaget's philosophy of science: Piaget vs. kuhn on scientific progress. *Theory & Psychology*, 16(2):203–224.

Alan Tygel and Rosana Kirsh. 2016. Contributions of Paulo Freire for a critical data literacy: A popular education approach. *The Journal of Community Informatics*, 12(3).

Jijo Pulickiyil Ulahannan, Nikhil Narayanan, Nishad Thalhath, Prem Prabhakaran, Sreekanth Chaliyeduth, Sooraj P Suresh, Musfir Mohammed, E Rajeevan, Sindhu Joseph, Akhil Balakrishnan, Jeevan Uthaman, Manoj Karingamadathil, Sunil Thonikkuzhiyil Thomas, Unnikrishnan Sureshkumar, Shabeesh Balan, Neetha Nanoth Vellichirammal, and the Collective for Open Data Distribution-Keralam (CODD-K) consortium. 2020. A citizen science initiative for open data and visualization of COVID-19 outbreak in Kerala, India. *Journal of the American Medical Informatics Association*, 27(12):1913–1920.

Georg Von Krogh and Eric Von Hippel. 2006. The promise of research on open source software. *Management science*, 52(7):975–983.

L. S. Vygotsky. 1978. *Mind in Society*. Harvard University Press.

April Yi Wang, Zihan Wu, Christopher Brooks, and Steve Oney. 2020. Callisto: Capturing the" why" by connecting conversations with computational narratives. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

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

Kailu Wang, Eliza Lai-Yi Wong, Kin-Fai Ho, Annie Wai-Ling Cheung, Peter Sen-Yung Yau, Dong Dong, Samuel Yeung-Shan Wong, and Eng-Kiong Yeoh. 2021. Change of Willingness to Accept COVID-19 Vaccine and Reasons of Vaccine Hesitancy of Working People at Different Waves of Local Epidemic in Hong Kong, China: Repeated Cross-Sectional Surveys. *Vaccines*, 9(1):62.

Ruotong Wang, Ruijia Cheng, Denae Ford, and Thomas Zimmermann. 2023. Investigating and designing for trust in ai-powered code generation tools.

Katherine E. Warren and Leana S. Wen. 2016. Measles, social media and surveillance in Baltimore City. *Journal of Public Health*, page jphm;fdw076v2.

David Weintrop, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. 2016. Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1):127–147.

Etienne Wenger, Richard Arnold McDermott, and William Snyder. 2002. *Cultivating communities of practice: A guide to managing knowledge*. Harvard business press.

Jevin West and Jason Portenoy. 2016. The Data Gold Rush in Higher Education. In Cassidy Sugimoto, Hamid R. Ekbia, and Michael Mattioli, editors, *Big Data Is Not a Monolith*, Information Policy. MIT Press.

Michelle Wilkerson, William Finzer, Tim Erickson, and Damaris Hernandez. 2021. Reflective Data Storytelling for Youth: The CODAP Story Builder. In *Interaction Design and Children*, IDC '21, pages 503–507, Athens, Greece. Association for Computing Machinery.

Leland Wilkinson. 2005. *The Grammar of Graphics*. Statistics and Computing. Springer New York, New York, NY.

Jeannette M. Wing. 2006. Computational Thinking. *Communications of the ACM*, 49(3):33–35.

Jeannette M Wing. 2008. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3717–3725.

David Wolber, Hal Abelson, Ellen Spertus, and Liz Looney. 2011. *App Inventor*. O'Reilly, Sebastopol, CA.

Annika Wolff, Michel Wermelinger, and Marian Petre. 2019. Exploring design principles for data literacy activities to support children's inquiries from complex data. *International Journal of Human-Computer Studies*, 129:41–54.

Kanit Wongsuphasawat, Yang Liu, and Jeffrey Heer. 2019. Goals, Process, and Challenges of Exploratory Data Analysis: An Interview Study.

Anbang Xu and Brian Bailey. 2012a. What Do You Think?: A Case Study of Benefit, Expectation, and Interaction in a Large Online Critique Community. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work - CSCW '12*, page 295, Seattle, Washington, USA. ACM Press.

Anbang Xu and Brian Bailey. 2012b. What do you think? a case study of benefit, expectation, and interaction in a large online critique community. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, page 295–304, New York, NY, USA. Association for Computing Machinery.

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

Seungwon Yang, Carlotta Domeniconi, Matt Revelle, Mack Sweeney, Ben U. Gelman, Chris Beckley, and Aditya Johri. 2015a. Uncovering Trajectories of Informal Learning in Large Online Communities of Creators. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale (L@S '15)*, pages 131–140, New York, NY. ACM.

Seungwon Yang, Carlotta Domeniconi, Matt Revelle, Mack Sweeney, Ben U Gelman, Chris Beckley, and Aditya Johri. 2015b. Uncovering trajectories of informal learning in large online communities of creators. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 131–140.

Yang Yang, Pei-Yu Chen, and Paul Pavlou. 2009. Open innovation: An empirical study of online contests. *ICIS 2009 Proceedings*, page 13.

Yu-Chun (Grace) Yen, Steven P. Dow, Elizabeth Gerber, and Brian P. Bailey. 2016a. Social Network, Web Forum, or Task Market?: Comparing Different Crowd Genres for Design Feedback Exchange. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS '16*, pages 773–784, Brisbane, QLD, Australia. ACM Press.

Yu-Chun (Grace) Yen, Steven P. Dow, Elizabeth Gerber, and Brian P. Bailey. 2016b. Social network, web forum, or task market? comparing different crowd genres for design feedback exchange. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS '16, page 773–784, New York, NY, USA. Association for Computing Machinery.

Lixiu Yu and Jeffrey V. Nickerson. 2011a. Cooks or Cobblers?: Crowd Creativity through Combination. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, CHI '11, pages 1393–1402, New York, NY, USA. ACM.

Lixiu Yu and Jeffrey V. Nickerson. 2011b. Cooks or cobblers? crowd creativity through combination. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 1393–1402, New York, NY, USA. Association for Computing Machinery.

Alexey Zagalsky, Carlos Gómez Teshima, Daniel M. German, Margaret-Anne Storey, and Germán Poo-Caamaño. 2016. How the r community creates and curates knowledge: A comparative study of stack

overflow and mailing lists. In *Proceedings of the 13th International Conference on Mining Software Repositories*, MSR '16, page 441–451, New York, NY, USA. Association for Computing Machinery.

Amy X. Zhang, Michael Muller, and Dakuo Wang. 2020. How do data science workers collaborate? roles, workflows, and tools. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW1).

Wenjun Zhou, Wangcheng Yan, and Xi Zhang. 2017. Collaboration for success in crowdsourced innovation projects: Knowledge creation, team diversity, and tacit coordination. In *Proceedings of the 50th Hawaii International Conference on System Sciences*.

Xinyi Zhou, Apurva Mulay, Emilio Ferrara, and Reza Zafarani. 2020. ReCOVery: A Multimodal Repository for COVID-19 News Credibility Research. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3205–3212, Virtual Event Ireland. ACM.

Appendix A

Chapter 3 Appendix

Figure A.1 shows the trend of percentage of games among all projects (not limited to those with variables or lists) over time as part of the analysis for H1 in §3.6.3. As shown in the figure, the overall percentage of games stayed consistent in the time period of our study.

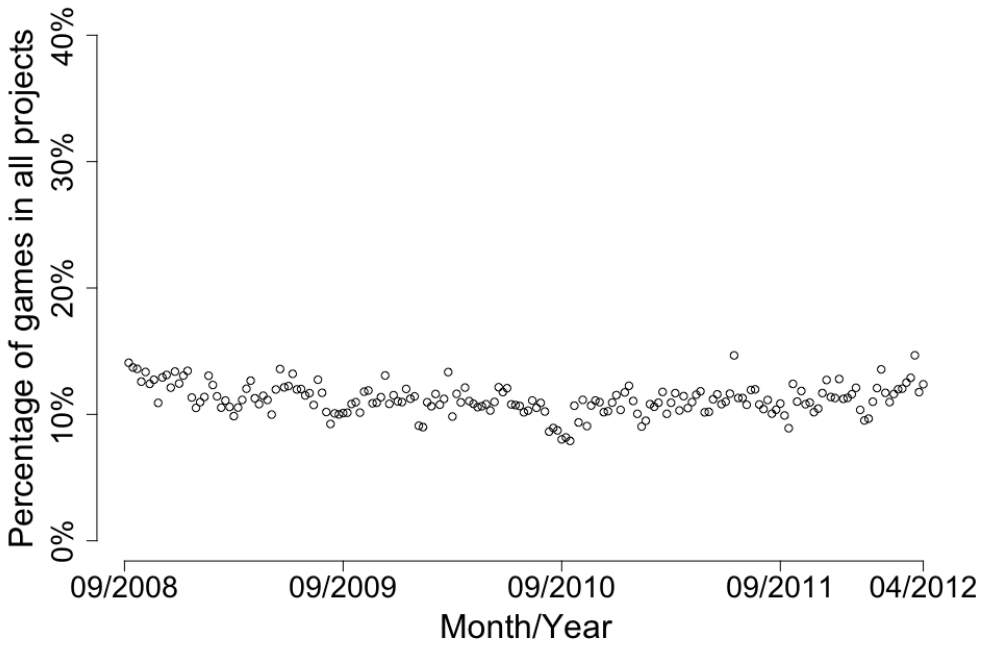


Figure A.1: Percentage of games among all projects in the community, per week, from September 2008 to April 2012.

To get a sense of whether the precision and recall of our strategy to identify games stayed consistent over time, we calculated the precision and recall in the first (09/02/2008-06/22/2010) and second half (06/22/2010-04/10/2012) of our study time period, for projects with lists and projects with variables respectively. The precisions were 0.83 and 0.88 for projects with lists and 0.95 and 0.83 for projects with variables. The recalls were 0.60 and 0.62 for projects with lists and 0.56 and 0.56 for projects with variables. This means the precision and recall largely stayed consistent.

Appendix B

Chapter 5 Appendix

Table B.1 shows the results of our validation analysis on keywords about data.

Keywords	Proportion of resulting tweets about data
“chart”	1.0
“charts”	0.7
“dashboard”	1.0
“dashboards”	1.0
“map”	0.9
“maps”	0.7
“plot”	0.2
“plots”	0.2
“viz”	0.0
“vis”	0.0
“visualization	1.0
“visualizations	1.0
“data”	1.0
“stats”	0.7
“statistics	0.7

Table B.1: Validation of keywords about data

Table B.2 contains the complete lists of hashtags used for collecting anti-vaccine and pro-vaccine tweets.

Pro-vaccine keywords	Anti-vaccine keywords	
vaccineswork	abolishbigpharma	exposebillgates
vaccineworks	noforcedflushots	vaccineharm
whyivax	antivaccine	forcedvaccines
vaccinessavelives	NoForcedVaccines	vaccineinjuries
vaccinessavelife	ArrestBillGates	Fuckvaccines
vaccinesaveslife	notomandatoryvaccines	vaccineinjury
vaccinesaveslives	betweenmeandmydoctor	idonotconsent
getvaccinated	NoVaccine	VaccinesAreNotTheAnswer
accepttobevaccinated	bigpharmafia	informedconsent
vaccinated	NoVaccineForMe	vaccinesarepoison
voicesforvaccine	bigpharmakills	learntherisk
ivax2protect	novaccinemandates	vaccinescause
fullyvaccinated	BillGatesBioTerrorist	medicalfreedom
fullyvaxxed	parentalrights	vaccineskill
vaxwithme	billgatesevil	medicalfreedomofchoice
thisisourshot	parentsoverpharma	vaxxed
doublevaccinated	BillGatesIsEvil	momsofunvaccinatedchildren
doublevaxxed	saynotovaccines	yeht
	billgatesisnotadoctor	mybodymychoice
	stopmandatoryvaccination	antivax
	billgatesvaccine	cdcvax
	syringeslaughter	naturalimmunity
	cdcfraud	thetruthaboutvaccines
	unvaccinated	vaccinesdontwork
	cdctruth	antivax
	v4vglobaldemo	cdcvax
	cdcwhistleblower	naturalimmunity
	vaccinationchoice	thetruthaboutvaccines
	covidvaccineispoison	
	VaccineAgenda	
	depopulation	
	vaccinedamage	
	DoctorsSpeakUp	
	vaccinefailure	
	educateb4uvax	
	vaccinefraud	

Table B.2: Keywords used to collect tweets.

Appendix C

A Quantitative Analysis of Legitimate Peripheral Participation in Scratch

This appendix contains the paper "Many Destinations, Many Pathways: A Quantitative Analysis of Legitimate Peripheral Participation in Scratch" that I published at the ACM SIGCHI Conference on Computer-Supported Cooperative Work & Social Computing in 2022 (CSCW'22).

Although informal online learning communities have proliferated over the last two decades, a fundamental question remains: What are the users of these communities expected to learn? Guided by the work of Etienne Wenger on communities of practice, we identify three distinct types of learning goals common to online informal learning communities: the development of domain skills, the development of identity as a community member, and the development of community-specific values and practices. Given these goals, what is the best way to support learning? Drawing from previous research in social computing, we ask how different types of legitimate peripheral participation by newcomers—contribution to core tasks, engagement with practice proxies, social bonding, and feedback exchange—may be associated with these three learning goals. Using data from the Scratch online community, we conduct a quantitative analysis to explore these questions. Our study contributes both theoretical insights and empirical evidence on how different types of learning occur in informal online environments.

C.1 Introduction

In the last two decades, we have witnessed the proliferation of online communities that seek to promote informal learning through unstructured activities and interactions with others. For example, social computing scholars have documented the way users work together to learn creative and technical animation skills in communities such as *NewGrounds* (Luther et al., 2009), writing and web development skills in fan communities such as *FanFiction.net* and *Archive of Our Own (AO3)* (Campbell et al., 2016a; Fiesler et al., 2017a), and programming skills in creative coding communities such as *Scratch* (Resnick et al., 2009b). These communities do not offer fixed lesson plans and offer little in the way of formal instruction. They are interest-driven, open to all, and free to join. Members learn at their own pace while pursuing their personal passions.

Skeptics of informal learning have pointed out that many users of such systems spend a substantial portion of their time on these websites commenting and socializing. How effective can a programming community be in supporting learning if its users constantly chat with each other while spending relatively little time writing or reading code? Proponents have responded that socialization promotes continued participation, a prerequisite to learning in informal settings where users are always at risk of not returning, and allows a collaborative process of learning and mentoring (Dasgupta and Hill, 2018b; Shorey et al., 2020b; Campbell et al., 2016a). Others have suggested that learning to socialize is an important learning outcome itself (Evans et al., 2017; Bruckman, 2005; Ito, 2013). An effective coding community could help its members develop important social skills in addition to technical ones (Brennan et al., 2011).

There remains a deep disagreement within the social computing community about how learning communities should be designed to maximize learning. We believe that resolving these conversations is difficult because researchers often do not fully answer the following questions: What do we expect members of informal learning communities to learn? What types of behavior should designers encourage and support for each learning goal?

Our work begins to explore answers to these questions. Guided by Lave et al. (1991)'s influential work on communities of practice (CoPs) and Wenger et al. (2002)'s more recent work on types of learning in CoPs, we describe three distinct categories of learning outcomes common to informal online learning communities: development of *domain* skills, development of *community* identity, and development of community

practices. Using these three outcomes, we explore how different types of legitimate peripheral participation (LPP) by newcomers—contribution to core tasks, engagement with practice proxies, social bonding, and feedback exchange—are associated with these different learning outcomes using data from the Scratch online community (Hill and Monroy-Hernández, 2017). We find that different types of participation are associated with learning outcomes differently. For example, making original projects as a newcomer is a good predictor of learning new computational concepts in the long term, while it is negatively associated with the formation of community identity and development of community practices.

Our paper makes several contributions. First, we believe that ours is the first study to provide a quantitative analysis of LPP and learning outcomes in a CoP. In doing so, our work suggests that users' early participation in an online community is associated with long-term learning outcomes—but the successful types of participation associated with different types of learning outcomes will differ. Second, we make an empirical contribution by analyzing 3 years of longitudinal data from Scratch. We believe that our work provides a template for future work that seeks to quantitatively investigate CoP theory in social computing in the context of a computational learning community. Finally, our work informs the design of new systems by pointing out potential ways to facilitate different newcomer participation patterns supporting a range of learning outcomes.

C.2 Background

C.2.1 Communities of practice

Over the last three decades, Jean Lave and Etienne Wenger's ideas around CoP have been among the most important theories used by social computing scholars to understand how learning happens in communities. The term *community of practice* has been widely used in social computing scholarship to describe groups of people learning from each other while working toward a common interest or goal (e.g., Shrestha et al., 2021; Marlow and Dabbish, 2014a; Kou et al., 2018; Gilbert, 2016). First introduced by Lave et al. (1991) in 1991, CoP theory is grounded in ethnographic observation of apprenticeship relationships in communities of Liberian tailors, Mayan midwives, US Navy quartermasters, nondrinking alcoholics, and US supermarket meat cutters.

Learning in CoPs is described as occurring through LPP, the phenomenon through which newcomers begin to participate in a group by helping out with tasks that are easy and low-risk but still valuable and important. For example, a new midwife may begin by boiling water and cleaning scissors for other midwives. Through observing more experienced members performing more complex and higher-stake tasks—and by practicing themselves in various ways—novices move from the periphery to more central roles.

C.2.2 Applications of CoP theory in social computing research

Although created to explain learning through traditional apprenticeships offline, the HCI and CSCW communities have embraced the CoP framework. Today, it is one of the most influential and highly cited theories related to learning in social computing. CoP has been widely adopted by social computing scholars because it is a good match for informal and individualized forms of learning that occur in many online settings.

Despite being cited hundreds of times in social computing scholarship—and tens of thousands of times in general—there have been very few attempts to explore or test CoP theory quantitatively. We are not aware of any such attempts in social computing scholarship. Furthermore, the work of applying CoPs to online settings, where groups are often extremely porous and diffuse, has required new theoretical work and argumentation. For example, Gruzd et al. (2011) have argued that a single social media user can form a personal or "imaginary" CoP with other users in their network. Using a similar line of thinking, data scientists on Twitter who engage with the hashtag "#TidyTuesday" have been theorized to constitute a CoP because they share context, common interests, and a process of collaborative knowledge advancement and because they ask questions and interact with more experienced users (Shrestha et al., 2021).

A wide range of other examples of online CoPs that have been identified and studied in social computing includes design professionals on online critique platforms (Marlow and Dabbish, 2014a; Kou et al., 2018), a Facebook group of Airbnb hosts who learn about new features and hosting strategies from each other (Holikatti et al., 2019), and fan fiction authors who gather online to develop and maintain the fan fiction website Archive of Our Own (AO3) (Fiesler et al., 2017a). In these examples and others, online CoPs are generally described as occurring in *affinity spaces* where members with similar interests and identities contribute to a shared collection of knowledge in a distributed manner (Gee, 2005). An example of an online affinity space that is frequently described as a CoP is the Scratch online community—the empirical context

of our study (Monroy-Hernández, 2007).

C.2.3 Types of learning in CoPs

One limitation in Lave and Wenger's initial account is that it is vague about what exactly is being learned in a given CoP. This is an important omission because CoP members typically learn a range of different things as they become more experienced. For example, becoming a Mayan midwife in the Yucatan Peninsula involves much more than learning technical midwifery skills. It also involves learning about the Mayan midwifery community and the specific norms and values that shape midwifery in the Yucatan. In later work, Wenger et al. (2002) attempted to address this omission by identifying three types of learning in CoPs: learning about *domain*, *community*, and *practice*. We present each type of learning in the following paragraphs.

First, learning about a *domain* refers to the acquisition of knowledge and skills necessary for a person to carry out the core tasks at the heart of a CoP. Many scholars of online communities are interested in how communities use LPP to support learning about some domain of knowledge and domain learning is often assumed to be the only learning goal in a CoP. For midwives, learning about a domain involves gaining knowledge related to successfully delivering infants. Depending on the community, domain knowledge may involve skills related to computer programming (Resnick et al., 2009b; von Krogha et al., 2003), fan fiction writing (Campbell et al., 2016a), encyclopedia article editing (Halfaker et al., 2013b), and so on. In the specific context of computational learning communities such as Scratch, domain learning means learning computational concepts and related programming skills (Resnick et al., 2009b).

The second type of learning involves the development of identity as a member of the *community*. This involves developing relationships, affinities, and a sense of belonging. As learners are accepted by older members of the community, they gradually form an identity as a member of the community. The development of these relationships and affinities would play out similarly in various settings and typically involves the knowledge of other community members and the development of a sense of membership and commitment to the community (McMillan and Chavis, 1986).

Third and finally, learning a *practice* means assimilating "cultural artifacts, norms, and values" developed in the community over time (Barab and Duffy, 2000). By moving from peripheral to central forms of participation, learners develop by adjusting their social, work, and contribution style to match what is

accepted and appreciated by community members. The specific reasons why one would be seen as accepted and appreciated in a CoP are very community-specific. However, indicators of practice development will typically involve expressions of appreciation and respect from others. For example, for midwives, it might involve the authority to lead or manage other midwives (Lave et al., 1991). On Fanfiction.net, signs of practice development may involve high ratings and positive comments (Campbell et al., 2016a). On Scratch, it might involve "loves" (i.e., likes) given by other users (Brennan and Resnick, 2013).

C.2.4 How does LPP promote different types of learning in CoPs?

Just as CoPs promote multiple types of learning, they also allow multiple types of participation. Although many studies of online activity reduce behavior to unidimensional concepts like "engagement," there are many types of LPP that occur in CoPs. To identify different types of LPP relevant to online communities, we conducted a detailed search of the literature on CoPs in CSCW and social computing venues. We did not find any previous attempt to enumerate or classify different types of LPP in social computing. However, through our reading of the literature, we were able to identify four distinct types of LPP that are frequently discussed: contribution to core tasks, engagement with practice proxies (an important type of activity in CoP theory, which we will explain in detail below), the formation of social bonds, and feedback exchange. We do not claim that these four types are strictly mutually exclusive; nor do we claim that they form a comprehensive list. We merely offer them as examples of four different types of participation that capture some of the diversity of LPP. We discuss the limitations of our work based on our necessarily arbitrary identification of the types of LPP in §C.8.

The first type of LPP we identified, *contribution to core tasks*, refers to the work of newcomers toward a community's explicit goal. Examples of contributions to core tasks include editing articles on Wikipedia, submitting code in an open source software project, and creating programming projects on Scratch. The second type of LPP is engagement with *practice proxies*. The term practice proxies refers to activities that allow newcomers to observe and participate in the socially salient aspects of others' unfolding work practices (Mugar et al., 2014). For example, in CoPs for online citizen science, new users can learn to contribute by engaging with project documentation left by others (Mugar et al., 2014). The third type of LPP is *social bonding*. In Lave et al. (1991)'s study of an alcoholics social group, the most important

activities involved members forming interpersonal bonds and becoming friends. In online CoPs, social bonding is often enabled by social media features such as friending and following. The fourth form of LPP is participation in *feedback exchange*. Feedback exchange in online CoPs often takes place through public commenting on artifacts shared by community members.

Because participants in online groups learn different types of things, it stands to reason that the most effective forms of LPP for the promotion of learning might vary depending on the type of learning. In other words, not all forms of peripheral participation will be equally likely to help a newcomer develop in terms of every desirable learning outcome. For example, peripheral participation in socializing activities might help newcomers build knowledge about the community, but might not contribute to their knowledge of domain skills. Consequently, we might not expect that socializing in a coding-focused CoP would necessarily make one a better programmer. Similarly, contribution to the core task of programming may not help to build social knowledge about the community and its members. Indeed, given limited time and resources, support for one learning outcome might come at the expense of others. For example, a study of the Scratch community showed that socialization among members in the comment section of Scratch projects can sometimes drive discussion away from programming related topics (Shorey et al., 2020b).

Inspired by the original CoP literature that focused primarily on the experience of newcomers and their progression in the community (Lave et al., 1991), we specifically focus on the participation of newcomers and their long-term learning outcomes. Figure C.1 includes the four types of newcomer LPP we have described on the left and Wenger et al. (2002)'s three types of learning in CoPs on the right. We draw 12 left-to-right arrows that represent all possible direct pathways between our four types of LPP and Wenger et al. (2002)'s three learning outcomes. In the following sections, we explain our research questions (R1 to R3) and related work on how different types of LPP contribute to the three learning outcomes.

Domain

The social computing literature suggests that domain knowledge learning can be influenced by different types of LPP, although the specific effects remain unclear. For example, many studies suggest that by contributing to core tasks, newcomers can learn technical details and gain confidence. For example, new Wikipedia editors often start by making edits on topics that they are familiar with (Bryant et al., 2005). In

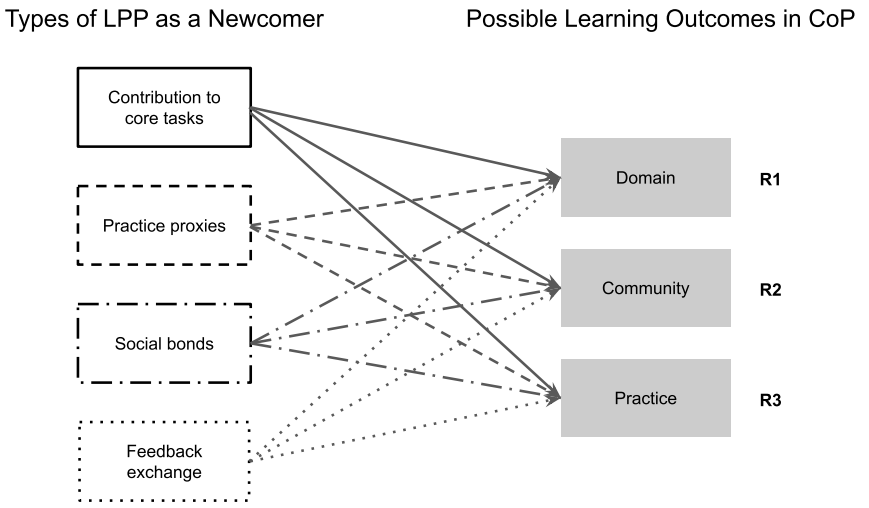


Figure C.1: Our research examines which type of newcomers' LPP is associated with which learning outcome in CoP.

the context of computational learning, novice programmers involved in open source projects often begin by reporting bugs and requesting features (von Krogha et al., 2003) and fanfiction authors learning to program contribute to the development of a fanfiction website by engineering very small features (Fiesler et al., 2017a). However, some also argue that contributing to core tasks can be stressful and overwhelming to newcomers, especially for those without much previous exposure to the domain.

Alternatively, newcomers can learn domain knowledge and skills with the help of practice proxies. For example, social media functionality that allows community members to view and download artifacts shared by others has been described as a method by which practice proxies can build domain knowledge among new designers (Marlow and Dabbish, 2014a). Similarly, new Wikipedia editors rely on practice proxies by reading existing pages and page histories before making their initial contributions to the encyclopedia as a way of learning where, when, and how to edit (Bryant et al., 2005). In the context of computational learning, the evidence of the relationship between engagement with practice proxies and domain learning is mixed. While research shows that Scratch users who create remixes of others' projects (a common practice proxy in computational learning) will learn more computational concepts as they continue to participate in

the community (Dasgupta et al., 2016a), they tend to display less innovation and originality in the programs they make (Hill and Monroy-Hernández, 2013).

Furthermore, while many computational learning platforms support socialization and feedback exchange, how these activities contribute to computational learning remains unclear. To explore how different types of LPP contribute to domain knowledge learning in CoPs, we ask the research question **R1:** How do different types of LPP affect domain knowledge learning?

Community

Qualitative evidence has shown that *engagement with practice proxies*, forming *social bonds*, and participating in *feedback exchange* can help newcomers develop an identity as a member of their *community*. Engagement with practice proxies can not only show newcomers how to master domain skills, but also provide newcomers with a sense of belonging. Lave et al. (1991) describe how novice meat cutters in a US supermarket who lacked practice proxies had a low rate of interaction with more experienced butchers and did not feel like they were participating in a community to learn skills. In the context of Scratch, many users engage in remixing with the sole purpose of broadcasting each other's work and being part of a social group (Hill and Monroy-Hernández, 2013). In the online CoP of R data scientists on Twitter, a large portion of newcomers join to build connections with others in the R community (Shrestha et al., 2021). Furthermore, social computing research has shown that newcomers with connections to others tend to remain active in the community longer (Kraut et al., 2011).

Previous research has indicated that newcomers' engagement in feedback exchange can also help users develop as community members. Feedback exchange in online CoPs has been described as a form of distributed mentoring in which mentoring relationships are not bounded by time, geographic space, and differences in expertise (Campbell et al., 2016a). Similarly, an experiment on reader engagement on Wikipedia has shown that newcomers showed higher long-term participation rates when asked to provide feedback on articles (Halfaker et al., 2013b). To obtain quantitative evidence for how different types of LPP support members to develop community identity, we explore the research question **R2:** How do different types of LPP affect the development of community identity?

Practice

Few papers in the social computing literature have directly studied how *practice* in a community—i.e., being able to understand and reproduce what is valued by CoP members—can be supported. Some qualitative studies have implied that feedback exchange could help members of online CoPs learn community practices. For example, in forums dedicated to online dating, users offer each other comments and advice on self-presentation to understand how others use dating sites and what behaviors are considered appropriate (Masden and Edwards, 2015). Similarly, new Airbnb hosts critique each other's drafts of hosting descriptions to recognize effective hosting styles (Holikatti et al., 2019) and young UX enthusiasts comment on others' posts to help each other learn how to make an appropriate networking message in the community (Kou et al., 2018). To further explore if and how different types of LPP can help learners learn community practices differently, we ask the research question **R3:** How do different types of LPP affect the learning of community-specific values and practices?

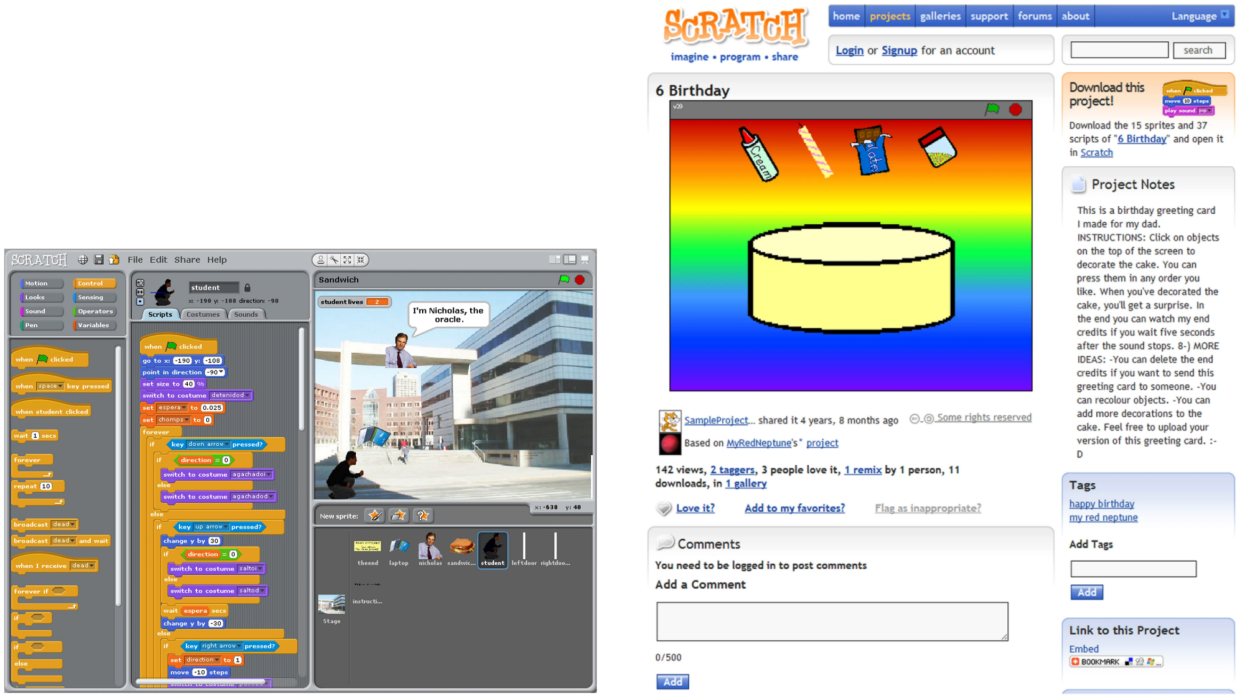
C.3 Empirical Setting: Scratch Community

We conducted our study using data from the Scratch community. Scratch was designed to support young people in learning to program using the Scratch programming language—a visual block-based language designed for children to learn basic programming (Resnick et al., 2009b; Roque et al., 2012). Scratch was designed based on constructionist learning principles (Papert, 1980b; Resnick et al., 1996b). Programming primitives are represented by visual blocks that control the behavior of graphical objects on the screen called sprites. As shown in Figure C.2a, users can drag and drop blocks together to build programs.

Although the Scratch programming language was designed for children to use on their own computers, the language has been integrated into a vibrant online community since 2007. Users in this community can view others' project and share their programming projects with others (Monroy-Hernández, 2007).[1] As of July 2021, the Scratch online community has over 73 million registered users and over 79 million shared projects that span a diverse range of genres and themes. The large majority of Scratch users are between the ages of 8 and 16 years and the average age for new contributors is approximately 12 years.[2]

[1] `https://scratch.mit.edu/` (permalink: `https://perma.cc/BRQ9-M3D6`)

[2] All statistics about Scratch community activity and users are taken from the public information on: `https://scratch.`



(a) Example of Scratch code.

(b) Social features and activities around the project.

Figure C.2: Historical interfaces of Scratch and social features at the time when data used in our analysis were collected. Images obtained from Hill and Monroy-Hernández (2017).

We chose Scratch for our study because it has been the center of several influential studies on informal learning in social computing and has been described as a CoP or site for situated learning in a range of previous studies (Dasgupta et al., 2016a; Hautea et al., 2017b; Dasgupta and Hill, 2018b). Beyond its importance in social computing scholarship, Scratch is one of the largest online communities for children learning to code. Critically for our analysis, Scratch supports a variety of learning outcomes and forms of participation that allow us to test all three of our research questions and test the hypotheses associated with each arrow in Figure C.1. Scratch users have access to practice proxies in the form of affordances that support users in viewing, editing, and building on others' code by downloading and remixing each other's projects (Monroy-Hernández, 2007). Previous studies have found evidence that remixing others' projects is associated with measures of learning about programming (Dasgupta et al., 2016a). Scratch also supports users to build social connections by "friending" other users to form social networks and a shared sense of community (Brennan et al., 2011; Brennan and Resnick, 2013). Scratch users can participate in feedback

mit.edu/statistics/ (permalink: https://perma.cc/4JEN-DYJD)

exchange by commenting on each others' projects, as shown in Figure C.2b. For example, previous research has revealed collaborative debugging activities in project comments (Shorey et al., 2020b). Finally, users can demonstrate social support by "loving" (upvoting) and "favoriting" (bookmarking) projects of others. We describe how these features map to our measure in the following section.

C.4 Data and Measures

Our data are drawn from publicly available *Scratch Research Dataset* (SRD), which includes comprehensive public data from the first 5 years of activity in Scratch between 2007 and 2012 (Hill and Monroy-Hernández, 2017). We further restrict our analysis to projects created between 2 July 2009 and 10 April 2012 because both programming and social affordances in the site were consistent during this period.[3] Our analysis relies on the tables *projects*, *users*, *pcomments*, *gcomments*, *project_blocks*, *favorites*, *lovers*, *viewers*, and *friends* from the SRD. Although this dataset is almost 10 years old, limiting ourselves to data from this period means that we can be more sure that the data are generated consistently and that others can reproduce our analysis using public data. To further support reproducibility, we have made the complete analytical code used in our analysis available.[4] As in many other large online communities, Scratch users frequently register and participate only briefly before becoming inactive (Crowston and Fagnot, 2018). Because we are interested in how users' activities as newcomers predict long-term outcomes, we created a user-level dataset with 121,149 users who were active for at least one day after their initial registration during our period of observation.

For all of our research questions, we attempt to measure activity during two periods: users' time spent as newcomers and their time as established users. Following a series of previous studies on Scratch, we define the "newcomer period" of Scratch users as consisting of the first 14 days after each user creates their account on Scratch. Like participants in most online communities, Scratch users' transition from newcomers to established members is fuzzy and variable. From our own participation in the Scratch community, we sense that nearly all users would be considered newcomers on their first several days, but most users would no longer be considered newcomers after more than a month of activity. We choose 14 days because it falls between this range and because it is embedded in one of Scratch's key platform policies in that "Scratcher"

[3] `https://en.scratch-wiki.info/wiki/Scratch_1.4` (permalink: `https://perma.cc/F42G-V6VV`)
[4] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/0VXEMB

status is only granted to users who have been on the platform for at least 14 days.[5] Within Scratch, users with Scratcher status are typically not considered newcomers by other community members and have access to more advanced programming features, such as the Scratch Cloud Variables (Dasgupta and Hill, 2018b). Other social computing studies have also defined the first 14-day period as the newcomer period and use user behaviors in this period to predict subsequent activities (Burke et al., 2009). Because this choice of 14 days is necessarily arbitrary, we also repeated our analyses with 2, 7, and 30 days as the length of the newcomer period in a series of robustness checks. All these analyses lead to similar results and conclusions. We include Table C.1 to give a sense of the distribution of activity in the first 14 days in our sample. A pattern—common in online community data from many sources—is that our measures are heavily right-skewed. Although some users are very active, the median number for most measures is 0.

Measure	Mean	St. Dev.	Median	Range
Original projects made in the first 14 days	2.200	4.515	1	[0, 222]
Remixes made in the first 14 days	0.656	2.219	0	[0, 130]
Friends made in the first 14 days	1.269	9.115	0	[0, 1,487]
Comments made in the first 14 days	4.516	26.640	0	[0, 2,450]
Friends received in the first 14 days	0.832	2.877	0	[0, 206]
Loves received in the first 14 days	0.492	2.668	0	[0, 261]
Views received in the first 14 days	11.133	34.935	3	[0, 4, 161]
Remixes received in the first 14 days	0.111	1.528	0	[0, 446]
Comments received in the first 14 days	2.706	16.079	0	[0, 1,418]
Favorites received in the first 14 days	0.271	1.529	0	[0, 160]
CT concepts displayed in the first 14 days	2.886	2.364	3	[0, 6]
Total active duration (in days)	110.656	171.079	36	[1, 1,002]
Total new CT concepts	4.281	1.981	5	[0, 6]

Table C.1: Descriptive statistics for a range of measures of user activity in Scratch including all activities that factor into our analysis.

Because most of the variation in the variables is between 0 and 1, we transform most of these variables into simple dichotomous measures for analysis. Each of the measures used in our regression analysis is shown in Table C.2. Three variables—*Stayed?*, *New_CT_concepts?*, and *Received_new_loves?*—are our binary outcome variables. The rest are independent variables and controls. All of these are drawn directly from the SRD or computed through merging and aggregating user-level activities across tables in ways that are self-explanatory and well documented in our code.

[5] `https://en.scratch-wiki.info/wiki/Scratcher` (permalink: `https://perma.cc/S4L6-XHH8`)

Variable	Mean	St. Dev.	Median	Range	Type
Independent variables:					
Made_original_project?	0.710	-	1	{0,1}	binary
Made_remix?	0.271	-	0	{0,1}	binary
Made_friend?	0.247	-	0	{0,1}	binary
Made_comment?	0.344	-	0	{0,1}	binary
Control variables:					
Has_been_friended?	0.271	-	0	{0,1}	binary
Has_been_loved?	0.170	-	0	{0,1}	binary
Has_been_viewed?	0.725	-	1	{0,1}	binary
Has_been_remixed?	0.057	-	0	{0,1}	binary
Has_been_commented?	0.329	-	0	{0,1}	binary
Has_been_favorited?	0.116	-	0	{0,1}	binary
CT_concepts	2.886	2.364	3	[0,6]	count
Outcome variables:					
Stayed?	0.678	-	1	{0,1}	binary
New_CT_concepts?	0.396	-	0	{0,1}	binary
Received_new_loves?	0.044	-	0	{0,1}	binary

Table C.2: Distribution of variables used for regressions.

The most important exception to this is our measures related to computational thinking (CT) concepts. CT is defined as "the thought processes involved in the formulation of problems and their solutions so that the solutions are represented in a form that can be carried out effectively by an information processing agent" (Wing, 2008). Referring to Brennan and Resnick (2012b)'s interpretation of CT and categorization of CT concepts, Scratch's designers designed programming blocks to embody specific CT concepts, including *loops*, *parallelism*, *events*, *conditionals*, *operations*, and *data* (Resnick et al., 2009b). Our measure of *CT concepts* captures the number of each distinct concept that a user displays in their projects made in their first 14 days. The specific CT concept-to-block mapping that we use is drawn from the study by Dasgupta et al. (2016a) and can be found in our appendix in Table C.4.

C.4.1 Measures

Dependent Variables

Because Scratch does not make any systematic attempt to measure learning, an analysis like ours must rely heavily on proxy measures of our outcomes. We constructed three outcome variables that correspond to our three research questions. The outcome variable for **R1** measures the *domain* being learned in Scratch—the computational skills. To construct a measure to capture the learning of computational skills, we follow the approach of previous quantitative studies on computational learning in Scratch. Our measure captures the size of the learners' repertoire in terms of the number of types of CT concepts a user has demonstrated in their projects (Scaffidi and Chambers, 2012; Yang et al., 2015b; Dasgupta et al., 2016a; Dasgupta and Hill, 2018b). We construct an outcome measure to capture the learning of computational skills: *New_CT_concepts?*, a binary variable of whether the user uses any new CT concepts in their projects after the first 14 days.

For the outcome variable for **R2**, our proxy measure for *community* membership is the duration of a user staying active in the community. Although this is not a direct measure of learning, CoP theory suggests that users who are more integrated into a community will stay longer. We measure this outcome with a binary variable, *Stayed?*, that captures whether users will participate in any recorded community activities after 14 days. In this case, activity can include sharing original or remixed projects, posting comments, favoriting, and/or friending.

For **R3**, we want to test how participation in feedback exchange contributes to learning community-specific *practices* and values. Because it is difficult to directly measure users' learning of community values, we explore H3 using a proxy that seeks to measure the Scratch community's positive reaction to projects made by a user. Following previous work by Hill and Monroy-Hernández (2013), we measure the reaction of the Scratch community to a user's projects as the number of loves received by that user. Specifically, we construct the measure for the outcome variable in R3 as *Received_new_love?*, a binary variable of whether the user received any loves on projects created after their first 14 days.

Independent Variables

We construct four independent variables to measure the four types of LPP that we describe in §C.2.4. First, we construct a measure for the LPP of contribution to core tasks. Since the Scratch community is designed

to help children learn computer programming, the core task in Scratch is to write code and share original projects (Resnick et al., 2009b). Therefore, we construct a binary measure, *Made_original_project?*, that captures whether a user has shared an original project in their first 14 days. Second, we construct a measure for engagement with practice proxies. Drawing from previous literature on LPP in Scratch, we choose to operationalize practice proxies as *Made_remix?*, a binary variable of whether the user remixed in their first 14 days. Third, we construct a measure for social bonding. The most explicit way to form social bonds in Scratch is by adding other users as "friends." In Scratch, friending means that a user has followed another user and will receive notifications when they post new projects.[6] Because we want to measure whether a user is actively participating in forming social bonds, we construct a binary variable, *Made_friend?*, which captures whether a user has friended at least one other user in their first 14 days. Fourth, we constructed a measure for newcomers' participation in feedback exchange. In Scratch, feedback is exchanged mainly in the form of comments (Shorey et al., 2020b). There are two types of comments in Scratch: comments on projects and on "galleries." We sum these two types of comments into the total number of comments posted and received by each user. Because we want to ensure that we are measuring whether a user is actively engaged in feedback exchange, we construct a binary variable, *Made_comment?*, which captures whether a user has posted a comment in their first 14 days.

C.4.2 Control variables

Based on previous work, we add a series of control variables that capture reasons that users might differ in their learning outcomes beyond differences in the four types of LPP we identify. For example, because previous literature has suggested that social support can affect newcomers' learning (Burke et al., 2009; Kraut et al., 2011; Ford et al., 2016), we control for incoming social approval on projects, constructed as four binary variables—*Has_been_loved?*, *Has_been_favorited?*, *Has_been_viewed?*, and *Has_been_remixed?*—which measure whether a user received loves, favorites, views, and remixes on their projects, respectively, in their first 14 days on Scratch. We also construct two binary variables, *Has_been_commented?* and *Has_been_friended?*, to control for the effect of passively received feedback and social bonds. Because the programming experience of users before joining the Scratch community may affect users' learning of CT

[6] `https://en.scratch-wiki.info/wiki/Friend` (permalink: `https://perma.cc/Q8T9-UPBB`)

concepts and the reception of their projects within Scratch, we also included a control variable, *CT_concepts* that measures the total number of unique CT concepts used by a user in their first 14 days on Scratch. This control is only relevant in R1 and R3.

C.5 Analytic Plan

To answer R1 using the binary outcome variable *New_CT_concepts?*, we fit a logistic regression model on the dataset of 121,149 users using the GLM function in R.[7] Because the distribution of the count control variable *CT_concepts* is right-skewed (i.e., most users used 0 CT concepts), we use a started log transformation in all the models involving this control (i.e., $\log(x+1)$). Because our outcomes and all other measures are binary, our model is nonparametric, except for our *CT_concepts* variable. Our formal model for (M1) is as follows:

$$\log\left(\frac{\hat{p}(New\_CT\_concepts?)}{1-\hat{p}(New\_CT\_concepts?)}\right) = \beta_0 + \beta_1 Made\_original\_project? + \beta_2 Made\_remix? + \beta_3 Made\_friend? + \beta_4 Made\_comment? + \beta_5 Has\_been\_friended? + \beta_6 Has\_been\_loved? + \beta_7 Has\_been\_viewed? + \beta_8 Has\_been\_remixed? + \beta_9 Has\_been\_commented? + \beta_{10} Has\_been\_favorited? + \beta_{11} \log(CT\_concepts + 1)$$

To answer R2 with the binary outcome variable *Stayed?*, we fit a logistic regression model (M2) that is identical to the model in Equation C.5, except that it excludes the logarithmic-transformed variable *CT_concepts* and the associated parameter β_{11}. The dependent variable is as follows:

$$\log\left(\frac{\hat{p}(Stayed?)}{1-\hat{p}(Stayed?)}\right)$$

To answer R3 with the binary outcome variable *Received_new_loves?*, we fit a logistic regression model on the full dataset. Our formal model (M3) is the same as that in Equation C.5 but with the following dependent variable:

$$\log\left(\frac{\hat{p}(Received\_new\_loves?)}{1-\hat{p}(Received\_new\_loves?)}\right)$$

C.6 Results

As the first step in our analysis, we construct a series of bivariate graphs to show differences in the distributions of our outcomes across strata representing our key independent variables. These plots are shown in

[7] `https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/glm` (permalink: `https://perma.cc/EP9Q-MS8D`)

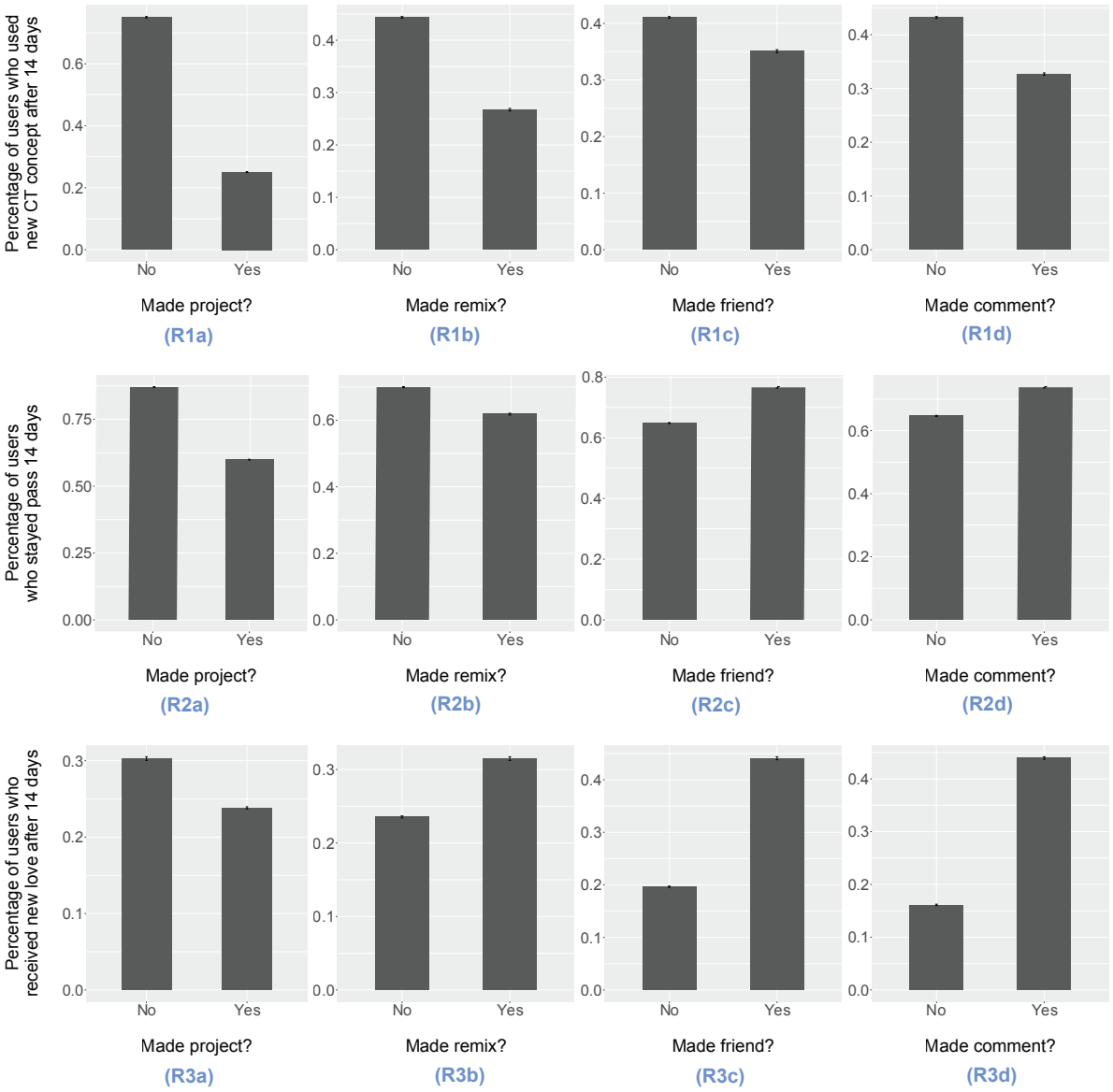


Figure C.3: Bivariate plots that show the differences in the distributions of outcome variables across strata of our dataset that reflect the independent variables associated with each of our research questions.

Figure C.3. In general, we find large differences in answers to all three of our research questions. Of course, these plots are exploratory; the observed relationships could be driven by any number of confounders. As a result, we explore our research questions using the full array of controls in our regressions.

The regression results for all three research questions are shown in Table C.3. Because interpreting marginal effects from regression models with many covariates can be challenging, especially in the case of

	New CT concept? (M1)	*Stayed?* (M2)	*Received new love?* (M3)
(Intercept)	1.47^{***}	2.27^{***}	-1.16^{***}
	(0.01)	(0.02)	(0.01)
Made_original_project?	0.44^{***}	-0.63^{***}	-0.71^{***}
	(0.03)	(0.02)	(0.04)
Made_remix?	-0.30^{***}	-0.24^{***}	0.00
	(0.02)	(0.02)	(0.02)
Made_friend?	0.01	0.34^{***}	0.28^{***}
	(0.02)	(0.02)	(0.02)
Made_comment?	0.00	0.48^{***}	0.95^{***}
	(0.02)	(0.02)	(0.02)
Has_been_friended?	0.20^{***}	0.38^{***}	0.42^{***}
	(0.02)	(0.02)	(0.02)
Has_been_loved?	-0.02	0.19^{***}	0.70^{***}
	(0.03)	(0.02)	(0.02)
Has_been_viewed?	-1.25^{***}	-1.70^{***}	-0.76^{***}
	(0.02)	(0.03)	(0.03)
Has_been_remixed?	0.04	0.12^{***}	0.31^{***}
	(0.03)	(0.03)	(0.03)
Has_been_commented?	-0.13^{***}	0.00	0.26^{***}
	(0.02)	(0.02)	(0.02)
Has_been_favorited?	-0.04	0.17^{***}	0.41^{***}
	(0.03)	(0.03)	(0.02)
log1p(*CT_concepts*)	-1.19^{***}	N/A	0.17^{***}
	(0.02)		(0.02)
AIC	125313.82	133601.48	119286.65
BIC	125430.28	133708.23	119403.11
Log Likelihood	-62644.91	-66789.74	-59631.33
Deviance	125289.82	133579.48	119262.65
McFadden's pseudo R-squared	0.23	0.12	0.14
Num. obs.	121149	121149	121149

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table C.3: Logistic regression models that predict the probability that a user uses a new CT concept (M1), stays active (M2), and receives loves on new projects (M3) created after their first 14 days on the community. The models are fit to the user-level dataset that includes aggregated activities of $121,149$ users.

GLM models such as logistic regression, we present a series of plots of model-predicted values in Figure C.4. We do so by identifying the median values for each of our control variables (shown in Table C.2) and then generating model predicted values for two prototypical users that vary only in terms of our key independent variables—e.g., a user who has (or has not) remixed, and so on. Each of the key independent

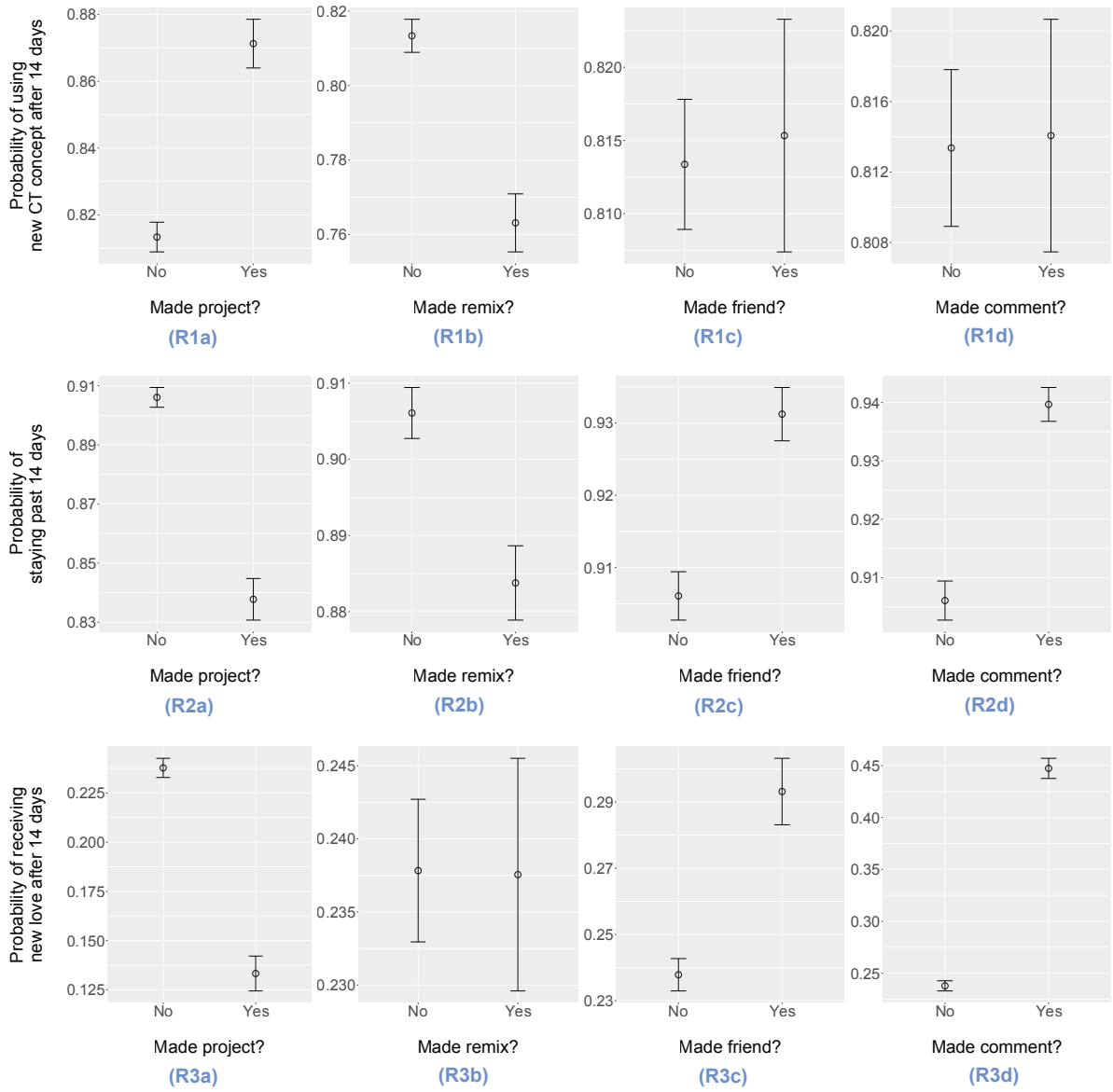


Figure C.4: Plots of model predicted probabilities corresponding to each of our research questions. Each figure shows the model predicted probabilities for two prototypical users who have median values for each of our control variables; these users vary only in terms of the key independent variable in the corresponding hypothesis. Error bars reflect the marginal effects of our key independent variable ($1.96 \times$ SE).

variables not being visualized is also held at the sample medians.

C.6.1 Domain

For R1, we found that making original projects as a newcomer is positively associated with the usage of new computational concepts in the long term, whereas remixing as a newcomer has the opposite effect. As shown in Table C.3, the odds that a user who created an original project in their first 14 days on Scratch would use new CT concepts afterward are 1.55 times the odds that an otherwise similar user who had not created an original project would do so ($\beta = 0.44$; SE $= 0.03$; $p < 0.001$). As seen in Figure C.4 (R1a), our model predicts that approximately 81% of the prototypical users who had not made original projects in their first 14 days would use new CT concepts afterward, whereas approximately 87% of otherwise similar users who had made original projects would. Furthermore, we find that the odds that a user who remixed in their first 14 days would use a new CT concept afterward are 0.74 times the odds of an otherwise similar user who had not done so ($\beta = -0.30$; SE $= 0.02$; $p < 0.001$). As seen in Figure C.4 (R1b), our model predicts that approximately 81% of prototypical users who had not remixed in their first 14 days would use new CT concepts afterward, whereas approximately 76% of the otherwise similar users who had remixed would. The effects of friending and commenting are not statistically significant.

C.6.2 Community

For R2, we found that making original projects and remixing as a newcomer are negatively associated with long-term stay in the community, whereas making friends and commenting as a newcomer were positively correlated with the probability of staying in the community for 14 days. The odds that a user who created an original project in their first 14 days in Scratch would stay in the community afterward are 0.53 times the odds that an otherwise similar user who had not created an original project would do so ($\beta = -0.63$; SE $= 0.02$; $p < 0.001$). As seen in Figure C.4 (R2a), our model predicts that approximately 91% of prototypical users who had not made original projects in the first 14 days will stay longer than 14 days, whereas only approximately 84% of otherwise similar users who had made original projects would stay. In terms of the effect of remixing, the odds that users who remixed others' projects in their first 14 days will stay longer than 14 days are 0.79 times the odds of a user who did not do so ($\beta = -0.24$; SE $= 0.01$; $p < 0.001$). As seen in Figure C.4 (R2b), our model predicts that more than 90% of prototypical users who did not remix in their first 14 days would stay in the community beyond 14 days, whereas approximately 88% of otherwise similar users who had remixed would. In contrast, the odds that users who friended others

in their first 14 days would stay past that time period are 1.40 times the odds that a user who did not friend anyone would ($\beta = 0.34$; SE $= 0.02$; $p < 0.001$). Figure C.4 (R2c) shows that our model predicts that approximately 90% of our prototypical users who had not friended anyone in their first 14 days would stay in the community past 14 days, whereas approximately 93% of otherwise similar users who had friended people would do so. For users who posted feedback in the community in their first 14 days, their odds of staying in Scratch after 14 days are 1.62 times the odds of users who did not ($\beta = 0.48$; SE $= 0.01$; $p < 0.001$). Figure C.4 (R2d) shows that our model predicts that only approximately 90% of prototypical users who had not posted any comments in their first 14 days would stay in the community past 14 days, whereas approximately 94% of otherwise similar users who had posted comments would.

C.6.3 Practice

For R3, we found that while making original projects as a newcomer is negatively associated with receiving loves from the community in the long term, making friends and commenting as a newcomer have a positive relationship with the outcome. Our model predicts that users who created an original project in their first 14 days in Scratch have 0.49 times the odds of receiving loves for their projects after 14 days compared with those who did not post feedback in the same newcomer period ($\beta = -0.71$; SE $= 0.04$; $p < 0.001$). As seen in Figure C.4 (R3a), our model predicts that approximately 24% of prototypical users who had not made original projects in their first 14 days will stay longer than 14 days, whereas only approximately 13% of otherwise similar users who had made original projects would. In contrast, the odds that users who friended others in their first 14 days would receive new loves are 1.32 times the odds that a user who did not friend anyone would do so ($\beta = 0.28$; SE $= 0.02$; $p < 0.001$). Figure C.4 (R3c) shows that our model predicts that approximately 24% of our prototypical users who had not friended anyone in the first 14 days would stay in the community past 14 days, whereas approximately 29% of otherwise similar users who had friended people would do so. In terms of feedback exchange, we find evidence that users who posted feedback in the community in their first 14 days would have 2.58 times the odds to receive loves for their projects after 14 days compared to those who did not post feedback in their newcomer period ($\beta = 0.95$; SE $= 0.02$; $p < 0.001$). As shown in Figure C.4 (R3d), the predicted probability of receiving loves for new projects created after 14 days is only approximately 25% for prototypical users who did not comment

during their initial period, but this value is close to 45% for those who did. The effect of remixing is not statistically significant.

C.7 Discussion

In this paper, we present a quantitative study with data from the Scratch community that investigates how newcomers' LPP contributes to different kinds of learning in CoP. Our paper makes a contribution to social computing theory by drawing in Wenger et al. (2002)'s three types of learning from the broader CoP literature: development of domain skills, development of identity as a community member, and development of community-specific values and practices. In our synthesis of the CoP literature in social computing, we identified four types of LPP common among newcomers (contribution to core tasks, engagement with practice proxies, social bonding, and feedback exchange) and formed three research questions related to understanding how the early participation of users in the community will predict learning outcomes in CoPs.

We contribute what we think is the first quantitative test of CoP theory in the context of social computing. Analytically speaking, our work tested 12 hypotheses that correspond to the 12 possible relationships between our independent and dependent variables (i.e., the lines in Figure C.1). Our work finds that most of these relationships are statistically significant, but that the signs and relative magnitudes of parameters associated with each type of LPP vary across the types of learning outcomes. At a very high level, our results provide concrete evidence that different types of LPP have different relationships with different important facets of learning in online CoPs. What is productive for some types of learning outcomes is unhelpful for others, and vice versa. We devote the remainder of our discussion to unpacking our specific results.

C.7.1 Supporting contribution to core tasks

Our study finds quantitative evidence for the belief that learners in a CoP will learn the domain by being part of the core tasks in the community. This finding is consistent with the constructionist design approach behind Scratch. In appears that by participating in epistemologically relevant project creation, learners can appropriate the abstract concepts behind their hands-on experiences (Papert, 1980b; Resnick et al., 1996b, 2009b).

That being said, contribution to core tasks is negatively associated with our measure of learning as

membership identity and community-specific practices. We speculate that because creating a programming project from scratch can be challenging for novices, some newcomers might feel discouraged and, as a result, turn away from the community. This is consistent with the results reported in previous literature, which indicate that it can be challenging for newcomers to follow community norms while making contributions (Halfaker et al., 2013a). Existing resources reflecting good practices are often composed by experienced users and contain contextual information that is not beginner-friendly (Marlow and Dabbish, 2014a; Fiesler et al., 2017a). These might be obstacles for newcomers looking to adopt community values.

Designers of the Scratch language have made enormous efforts to lower the bar of programming for novices (Resnick et al., 2009b). Still, making a new project, however simple, might be overwhelming for someone without programming experience. Future design of informal learning communities like Scratch could consider offering easier options for newcomers to engage in core tasks. For example, in some other online CoPs dedicated to software development, newcomers can participate in simple, collaborative, programming-related tasks such as reporting bugs (von Krogha et al., 2003). Scratch might try to direct newcomers toward types of newcomer-friendly programming tasks such as building on example templates, identifying bugs in collaborative projects, and helping more experienced users with a small part of their projects.

C.7.2 Supporting engagement with practice proxies

Our exploration shows that engagement with practice proxies is negatively associated with the development of domain knowledge and the formation of membership identity. Although previous work on Scratch shows that users who remixed projects containing a CT concept tended to use the same CT concept in their own projects (Dasgupta et al., 2016a), our findings suggest that this may not contribute to the expansion of computational knowledge and skills in general. Our results are consistent with previous findings on remixing, in which users tend to show less innovation and originality when building from the most frequently remixed programs (Hill and Monroy-Hernández, 2013).

More research needs to be conducted to understand exactly why remixing negatively impacts some dimensions of learning. One possible explanation is that newcomers may have trouble understanding specific choices in projects made by more experienced members and may not have the confidence needed to try and

build upon them. In Kaggle, novices face a similar challenge of understanding and reusing publicly shared code by experts due to incomplete documentation on dependencies, missing rationales, and lack of context (Cheng and Zachry, 2020b). In these cases, simply providing newcomers with practice proxies is likely not enough.

Possible solutions might include additional scaffolding by established community members to document their steps and reasoning processes in the first place. Wang et al. (2020) and Rule et al. (2018a) provide great guidelines and examples for such design in the context of computational learning. Informal learning communities should also offer newcomers the opportunity to ask questions to experienced members. Such a design should scaffold the asker to clearly communicate which step they are confused about in context and support answerers to effectively showcase their procedures and rationales. For example, in the context of Scratch, a synchronous Q&A function (e.g., an opt-in chat room similar to the one proposed by Ford et al. (2018b)) could be implemented in the remix feature so that newcomers could discuss the coding mechanisms with the original creators of the projects being remixed.

C.7.3 Supporting newcomer's socialization

Newcomers' social bonding activity is associated with the formation of community identity. Such socialization activities are also positively correlated with the development of community-specific practices. This finding can be read as a strong response to the skeptics of informal learning in online CoPs who argue that chatting and socialization will distract users. Our results complement other work that shows that socialization can be a legitimate pathway to learning what the community values (Cheng et al., 2022a). One possible interpretation is that social bonding can facilitate computational participation (Kafai, 2016)—learning through participating in socially situated contexts.

Better ways to socialize newcomers is a perennial topic in social computing. Our results suggest that success in these efforts can support a variety of learning outcomes. Proactive efforts to reach out to new users with comments are a useful way to jumpstart social bonding processes. Designers of informal online learning systems may consider intentionally helping newcomers socialize with each other and older members through community events or algorithmic matching mechanisms, for example, making the transition from identity-based connections to bond-based connections possible (Kraut et al., 2011).

C.7.4 Supporting feedback exchange

Feedback exchange is positively associated with our measures of learning about both community and practice. Complementing previous studies of collaborative debugging (Shorey et al., 2020b), our study offers quantitative evidence that socially oriented newcomer participation can contribute to community practices. Despite the promise of feedback exchange on learning, newcomers refrain from publicly offering feedback because they are self-conscious about their social status and are not confident in their expertise (Marlow and Dabbish, 2014a). Nonpublic or anonymous feedback systems might be established so that newcomers do not have to reveal themselves while interacting with community members (Ford et al., 2018b). Alternatively, scaffolds for composing discussion messages could be introduced so that newcomers can have some idea about what to say in comments and participate in feedback exchange more easily (Hui et al., 2018b; MacNeil et al., 2021).

C.8 Threats to Validity

A series of articles in the social computing literature have argued about whether deeply engaged and highly active users in online communities are "born and not made" (Huang et al., 2015; Panciera et al., 2009; Preece and Shneiderman, 2009). In its analytic structure, our work has quite a bit in common with *born versus made* studies by Panciera et al. (2009) and Huang et al. (2015). Both studies seek to make predictions about users' long-term behavior using measures of early activity. There are also important differences between these approaches and ours. Born/made studies typically look at users' very first engagement. Although we take a longer view in the construction of our measures, our results are not sensitive to this decision. Although we include a large body of control variables to address *ex ante* differences between users, we cannot fully rule out the explanation that users who are more active early on are systematically different from users who are not. In contrast, we might argue that our work points to an alternative explanation for early studies that suggest that users are "born and not made." Both our theoretical framework and empirical results suggest that early activity may not be best thought of as an indicator of existing differences in knowledge, commitment, and skills, but rather as a pathway through which users can develop.

Furthermore, we define the "newcomer" period as the first 14 days after a user creates an account.

Although this decision is informed by our experiences with Scratch and although robustness checks on user activities in the first 2, 7, and 30 days lead to consistent results, the choice of 14 days is arbitrary and fails to capture important nuance in the way that Scratch users understand their own status, and the status of others, within their community. We hope that future work can improve on this very limited measurement approach.

Another limitation of our work is that the four types of LPP that we identified are not a comprehensive list of all possible types of LPP in online communities, either in general or in Scratch in particular. As the result of our review of relevant social computing literature, our enumeration of these four types is put forward with humility and knowledge that is limited and incomplete. Although we know that our list is not comprehensive, we hope that our enumeration and measurement of four distinct types of LPP are sufficient to convey our high-level argument that there are different types of newcomer engagement that can lead to different learning pathways. Future research could explore other types of LPP and their impact on long-term learning outcomes.

Our work is also limited in that our measures are proxies rather than direct measures of the concepts of interest. This is particularly important to acknowledge in terms of our learning outcome measures. We rely on proxies because we simply do not have a way to evaluate actual learning in Scratch. For example, we use differences in incoming loves as a proxy for an individual's learning about community-specific practices and values. It is important to remember that receiving positive community reactions is an outcome of learning community practices, not a direct measure of learning. Another example is that we use a measure of whether a user stays in the community for longer than 14 days to proxy for a learner's development of membership identity. We rely on proxies because sociopsychological concepts are difficult to directly measure in informal settings like Scratch, where there is little or no ability to conduct tests or even communicate with users. Although we draw heavily from other work that has developed and evaluated proxy measures in Scratch, all of these attempts are risky and prone to noise, or worse (Dasgupta et al., 2016a; Dasgupta and Hill, 2018b). Although far from perfect, we believe that these proxies provide valuable insight into our research questions.

While our research questions describe causal relations, our regression analyses can only provide correlational evidence. As is often the case in cross-sectional analyses, these relationships might be due to variables that are correlated with, but not caused by, our predictors. We hope that design-based field experiments can be conducted in the future to identify causal relationships between specific types of newcomer LPP and the

specific kind of learning in CoPs. Until then, we offer our results as tentative evidence. Furthermore, our analysis focuses only on the main effect of the independent variables. In an earlier version of the analysis, we include two-way interactions of our independent variables as predictors in our models, and we include the results in Table C.6 in the appendix. Since understanding the interaction relationships between our independent variables is not the goal of this analysis and it is difficult to interpret the coefficients of interactions, we did not integrate these models into our main analysis. We urge future researchers to build on our work and explore the interactions and combination of different types of LPP and their contribution to different types of learning.

Our work is also limited in that all our outcomes are dichotomous. This is a choice we made for two reasons. First, it means that our analysis is at a lower risk of violating parametric modeling assumptions. Second, doing so also captures the vast majority of the variation in our outcomes. However, this choice means that our results only paint part of the picture. Although we show that sharing new projects and remixing early on are associated with a higher chance of using CT concepts later, we do not know if more remixing leads to more new CT concepts among users who continue. Although we leave this question for future work, we include an exploratory analysis in Table C.5 in our appendix that reproduces our three models with continuous measures of our dependent variables among the relatively small proportion of users for whom our outcomes were equal to 1. This is conceptually similar to a hurdle model. Although limited in several ways, these results suggest that our dichotomous measures, which are strong predictors of new CT concepts, staying active, and receiving loves, are not necessarily strong predictors of higher amounts of learning among users who have learned at all. We hope to refine and continue to explore the relationship between the types of newcomer LPP and the magnitude of learning outcomes in future studies.

Finally, our analysis focuses on Scratch as the single empirical setting. While Scratch is large, active, and supports a wide range of activities, it is also unique in many aspects: its user population mostly consists of young people, the community focuses on computational learning, and it contains affordances such as remixing and viewing others' full code. We do not know if our findings from the Scratch online community will generalize to other settings or population groups. In addition, we analyzed activities during a single time period, which was early in the Scratch community's lifetime. Since the affordances of Scratch have changed and the community itself has also grown since then, we cannot know how well our findings will

translate to Scratch today. We hope that our case study offers a framework for studying different types of LPP and learning in online CoPs. Future researchers could answer our research questions in other settings and compare their results with our findings.

C.9 Conclusion

In this study, we present a quantitative approach to engaging with CoP theory in an informal online learning context. Although previous studies in social computing have treated learning outcomes as a single dimension, we build on work by Wenger et al. (2002) to describe three distinct types of learning that can be supported in online CoPs: learning about the domain, learning about the community, and learning about the practice. We also identify four forms of LPP that are common in informal online learning contexts. We use historical data from Scratch to answer a series of research questions about whether certain types of newcomer LPP contribute to certain kinds of learning. Our results suggest that there is a range of possible pathways to a range of distinct learning outcomes.

Taking a broad exploratory approach instead of a more narrow hypothesis testing structure, we find evidence of relationships between types of LPP and learning outcomes that we believe are entirely untheorized in the CoP literature in social computing. For example, we find that newcomer socializing is associated with learning about the community and its practices. Our study offers theoretical and empirical contributions to social computing research on informal learning settings, as well as practical implications on how to best design informal online learning systems. In summary, our study suggests that online communities afford many possible destinations in learning and many pathways to each.

C.10 Appendix

CT concepts	Scratch Blocks
Loops	forever, foreverIf, repeat, repeatUntil
Parallelism	startHatTriggered, eventHatTriggered, keyHatTriggered, mouseHatTriggered
Events	eventHatTriggered, keyHatTriggered, mouseHatTriggered, bounceOffEdge, turnAwayFromEdge, touching, touchingColor, colorSees, mousePressed, keyPressed, isLoud, sensor, sensorPressed, distanceTo
Conditionals	waitUntil, foreverIf, if, ifElse, repeatUntil, bounceOffEdge, turnAwayFromEdge, touching, touchingColor, colorSees, mousePressed, keyPressed, isLoud, sensor, sensorPressed, lessThan, equalTo, greaterThan, and, or, not, listContains
Operators	lessThan, equalTo, greaterThan, and, or, not, add, subtract, multiply, divide, pickRandomFromTo, concatenateWith, letterOf, stringLength, mod, round, abs, sqrt, sin, cos, tan, asin, acos, atan, ln, log, e^, 10^
Data	setVarTo, changeVarBy, showVariable, hideVariable, readVariable, addToList, deleteLineOfList, insertAtOfList, setLineOfListTo, contentsOfList, getLineOfList, lineCountOfList, listContains

Table C.4: CT concepts mapping to blocks in the Scratch programming language, adopted from Dasgupta et al. (2016a). We use this mapping to construct our outcome variable for H1a and H1b about whether new CT concept is presented in projects created after a user's first 14 days in the community.

Variables	Total new CT concepts (M1)	Active duration (M2)	New loves per project (M3)
(Intercept)	4.65^{**}	4.81^{**}	0.15^{**}
	(0.01)	(0.01)	(0.00)
Made_original_project?	-0.09^{*}	-0.28^{**}	-0.15^{**}
	(0.04)	(0.02)	(0.01)
Made_remix?	0.02	-0.04^{*}	-0.04^{**}
	(0.02)	(0.01)	(0.00)
Made_friend?	-0.18^{**}	-0.03^{*}	0.02^{**}
	(0.02)	(0.01)	(0.00)
Made_comment?	-0.09^{**}	0.32^{**}	0.16^{**}
	(0.02)	(0.01)	(0.00)
Has_been_friended?	-0.06^{*}	0.09^{**}	0.08^{**}
	(0.02)	(0.01)	(0.00)
Has_been_loved?	-0.03	0.11^{**}	0.19^{**}
	(0.03)	(0.01)	(0.01)
Has_been_viewed?	0.00	0.06^{**}	0.00
	(0.03)	(0.02)	(0.01)
Has_been_remixed?	-0.02	0.11^{**}	0.05^{**}
	(0.03)	(0.02)	(0.01)
Has_been_commented?	0.01	0.14^{**}	0.01^{*}
	(0.02)	(0.01)	(0.00)
Has_been_favorited?	0.01	0.11^{**}	0.12^{**}
	(0.03)	(0.02)	(0.01)
log1p(CT_concepts)	-1.68^{**}	NA	0.03^{**}
	(0.02)	NA	(0.00)
R^2	0.47	NA	0.15
Adj. R^2	0.47	NA	0.15
AIC	NA	1140546.37	NA
BIC	NA	1140660.20	NA
Log Likelihood	NA	-570261.19	NA
Deviance	NA	116442.92	NA
McFadden's pseudo R-squared	0.16	0.01	0.14
Num. obs.	47952	97295	79963

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table C.5: Results from our early exploratory analysis based on continuous construct of outcome variables and subsets of our user level dataset used from the main analysis. M1_c tests H1, where *Total_new_CT_concepts* is a count variable of number of new CT concepts used in projects created after the first 14 days among the $47,952$ users who had used new CT concepts after 14 days. M2_c tests H2, where *Active_duration* is a count variable of the number of days that a user engages in any recorded community activities and is measured by the number of days between the the end of the users 14 day period and the last in which they are active. This is also constructed on the subset of $97,295$ users who were active after the first 14 days. M3_c tests H3, where *New_loves_per_project* is a continuous variable of average number of loves received by a user on projects created after the first 14 days, among the $79,963$ users who did create projects after 14 days. We chose to exclude these results from our main findings due to small sample size (< 10% of our original dataset), and truncation issues.

Variables	New CT concept? (M1)	Stayed? (M2)	Received new love? (M3)
(Intercept)	1.83^{***}	3.39^{***}	-1.04^{***}
	(0.02)	(0.04)	(0.02)
Made_original_project?	-0.79^{***}	-3.05^{***}	-1.63^{***}
	(0.04)	(0.05)	(0.06)
Made_remix?	-2.34^{***}	-3.59^{***}	-1.54^{***}
	(0.04)	(0.05)	(0.06)
Made_friend?	-0.08	0.30^{***}	0.21^{***}
	(0.05)	(0.07)	(0.04)
Made_comment?	0.03	0.52^{***}	1.01^{***}
	(0.04)	(0.06)	(0.03)
Has_been_friended?	0.17^{***}	0.35^{***}	0.40^{***}
	(0.02)	(0.02)	(0.02)
Has_been_loved?	-0.05	0.16^{***}	0.68^{***}
	(0.03)	(0.02)	(0.02)
Has_been_viewed?	-0.27^{***}	-0.35^{***}	0.14^{**}
	(0.03)	(0.03)	(0.05)
Has_been_remixed?	-0.01	0.08^{*}	0.29^{***}
	(0.03)	(0.03)	(0.03)
Has_been_commented?	-0.13^{***}	-0.01	0.25^{***}
	(0.02)	(0.02)	(0.02)
Has_been_favoriated?	-0.09^{**}	0.13^{***}	0.39^{***}
	(0.03)	(0.03)	(0.02)
log1p(CT_concepts)	-1.22^{***}	NA	0.13^{***}
	(0.02)	NA	(0.02)
Made_comment? × Made_remix?	0.22^{***}	0.17^{***}	0.11^{**}
	(0.04)	(0.04)	(0.04)
Made_comment? × Made_friend?	-0.01	0.11^{**}	-0.12^{***}
	(0.04)	(0.04)	(0.04)
Made_comment? × Made_original_project?	-0.20^{***}	-0.19^{***}	-0.10^{**}
	(0.04)	(0.06)	(0.04)
Made_remix? × Made_friend?	0.10^{*}	0.05	0.15^{***}
	(0.04)	(0.04)	(0.04)
Made_remix? × Made_original_project?	2.30^{***}	3.60^{***}	1.59^{***}
	(0.05)	(0.05)	(0.06)
Made_friend? × Made_original_project?	0.06	-0.06	0.13^{***}
	(0.04)	(0.06)	(0.04)
AIC	122375.61	128298.77	118346.10
BIC	122550.30	128463.75	118520.78
Log Likelihood	-61169.80	-64132.39	-59155.05
Deviance	122339.61	128264.77	118310.10
McFadden's pseudo R-squared	0.25	0.16	0.14
Num. obs.	121149	121149	121149

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table C.6: Logistic regression models that include two-way interaction between independent variables as predictors. Same as our main analysis, the models predict the likelihood that a user using new CT concept (M1), staying active (M2), and receiving loves on new projects (M3) created after their first 14 days in the community. Models are fit on the user level dataset including aggregated activities from 121149 users.