Skill Progression in Scratch Revisited

J. Nathan Matias
MIT Media Lab
Cambridge, MA, 02139
jnmatias@mit.edu

Sayamindu Dasgupta
MIT Media Lab
Cambridge, MA, 02139
sayamindu@media.mit.edu

Benjamin Mako Hill
University of Washington
Seattle, WA, 98195
makohill@uw.edu

ABSTRACT

This paper contributes to a growing body of work that attempts to measure informal learning online by revisiting two of the most surprising findings from a 2012 study on skill progression in Scratch by Scaffidi and Chambers: users tend to share decreasingly code-heavy projects over time; and users' projects trend toward using a less diverse range of code concepts. We revisit Scaffidi and Chambers's work in three ways: with a replication of their study using the full population of projects from which they sampled, a simulation study that replicates both their analytic and sampling methodology, and an alternative analysis that addresses several important threats. Our results suggest that the population estimates are opposite in sign to those presented in the original work.

ACM Classification Keywords

H.5.3 Information Interfaces and Presentation (e.g., HCI): Group and Organization Interfaces—*Evaluation/methodology*; K.3.2 Computers and Education: Computer and Information Science Education—*Information systems education*

Author Keywords

learning; online communities; computers and children; creativity support tools; replication

INTRODUCTION

In recent years, informal learning with digital tools has become increasingly prevalent in contexts including online communities, maker-spaces, and after-school computer clubs. One challenge in these informal learning environments is the measurement of learning outcomes, as there are rarely pre-specified learning goals and learning can occur in many ways. As these environments become increasingly widespread, it is important to not only formulate appropriate measures of learning but also validate and replicate findings. Moreover, within the larger sphere of human-computer interaction research, replication has been cited as important as it can help confirm previous findings and lead to better methodologies and measures [16].

CHI'16, May 07-12, 2016, San Jose, CA, USA
ACM 978-1-4503-3362-7/16/05.
http://dx.doi.org/10.1145/2858036.2858349

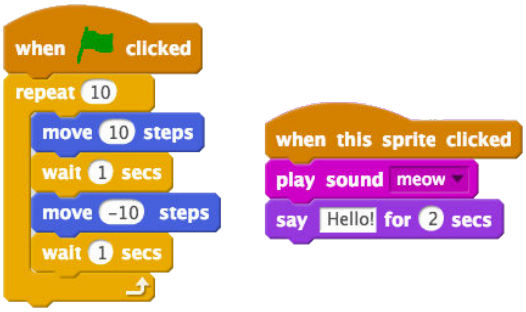

Figure 1. Example Scratch code with six blocks in the first "script" and three blocks in the second script. The first script makes a sprite move back and forth 10 times. The second script makes a sprite play a sound and show a speech bubble upon being clicked.

This paper re-examines part of the first quantitative evaluation of learning in the Scratch community, published in 2012 by Scaffidi and Chambers [14]. Scratch is a visual, block-based programming language designed for children aged 8–15, which young programmers have used to create a vast range of projects including animations, games, interactive stories, simulations, and science experiments [13]. In Scratch, programs are constructed by dragging and dropping visual *blocks*—similar to tokens in other programming languages—to define the behavior of on-screen graphical objects called *sprites* (Figure 1). The Scratch language is supported by a large online community launched in March 2007 and publicly announced in May 2007 [10], where creators can share their projects, comment on each others' work, and remix projects created by their peers.[1]

In particular, we revisit two of Scaffidi and Chambers's findings that ran counter to their hypotheses: that users tend to share decreasingly code-heavy projects over time; and that users' projects trend toward using a less diverse range of blocks. We revisit these findings as they are often cited in the literature on learning in informal online settings (e.g., [8, 12]), because their study uses a relatively small sample of 2,195 Scratch projects shared by 250 users from a population where a full dataset has been made available, and because the original results are surprising in terms of other research. In particular, these results seem to run counter to earlier work on Scratch suggesting that young learners in a computer club broadened their block usage and project count over time even without teacher intervention [9] as well as several qualitative studies that suggested that Scratch users grew in their learning of

[1] https://scratch.mit.edu

programming concepts over time [4, 11, 2]. Drawing from the same population as Scaffidi and Chambers but using different measures of learning, more recent work has also suggested that Scratch users tend to learn over time [17, 5].

This paper attempts to explore this apparent contradiction in three ways. First, we replicate Scaffidi and Chambers's relevant analyses with the full Scratch dataset. Second, we conduct a simulation study to estimate the chances of the earlier work's finding given their sample size. Third, we offer an alternative modeling approach that addresses important issues highlighted by our attempt at replication. By using the full dataset of 643,246 public non-remix projects shared by 138,321 users at approximately the time of data collection for the earlier work, we show that the actual relationships between these measures of skill and time are reversed in sign from Scaffidi and Chambers's original findings. By simulating small samples using their methodology, we are able to reproduce Scaffidi and Chambers's results in a way that suggests that the earlier findings are most likely the result of a small sample and an unlucky random draw. Finally, using the larger dataset, we are also able to present more detailed within-person estimates that support our new finding of a positive relationship. We argue that these new results help bring Scaffidi and Chambers's study in line with other research on learning and skill progression in similar settings.

SCAFFIDI AND CHAMBERS (2012)

Scaffidi and Chambers provide a detailed analysis of learning in Scratch, developing and testing a series of models of users' skill progression and levels of activity. In this paper, we focus on two of the paper's findings that contradict the authors' stated hypotheses. They find that there is a negative relationship between the *depth* and *breadth* of projects shared by users on Scratch and the time since users created their accounts.

The authors define depth as the "amount with which people used [Scratch's programming] features," which they measure as the total number of programming blocks used in a project (see Figure 1), excluding any measure of graphical elements or sounds. Breadth is defined as, "the range of different features people could use," measured as the total number of distinct categories of programming blocks used within a project. For their categorization, Scaffidi and Chambers group Scratch's 120 unique programming primitives into a set of 17 categories.

Scaffidi and Chambers sampled 250 users from the Scratch website by using a "Surprise Me" feature that displayed a random published project.[2] The authors then collected a sample of projects from each user's list of projects. Users' projects on the Scratch website are displayed in ascending order of time created and are paginated into groups of 15. Scaffidi and Chambers's dataset included each user's first project and one project selected randomly from each subsequent page. For example, if a user published 35 projects, the sample would include three projects: their first project, a project between 16 and 30, and a project between 30 and 35. Scaffidi and Chambers selected a total of 2,195 projects, of which 403 could not be analyzed due to version incompatibilities and errors. They also discarded one additional project from the remaining set as the creator of the project in question had simply copied a number of blocks repeatedly. In their study of project depth and breadth, they omitted remixed projects from their analysis.

To motivate their analyses, Scaffidi and Chambers hypothesized that "online Scratch users would demonstrate an increase in sophistication over time, with rising breadth and depth demonstrated by Scratch animations." To test this hypothesis, they created a measure of experience represented by the number of months that had elapsed since each project's creator shared their first project, rounded to the nearest month. They regressed this measure of creator experience on their measures of depth and breadth. Despite their expectations, they found a well-estimated negative relationship between depth and months ($\beta = -1.51, p < 0.01$) and a similarly well-estimated negative relationship between breadth and months ($\beta = -0.07, p < 0.01$).

Because learning happens within individuals, Scaffidi and Chambers also considered variation in a person-level version of their dataset. Among 145 users with projects shared in more than one month, they calculate the average depth and average breadth of a user's projects in their first and last months, testing the difference with two-tailed t-tests. In both cases, they found positive relationships between time on Scratch and breadth and depth, but a small sample size meant that these were poorly estimated. They could not reject the null hypothesis that there was no relationship.

REPLICATION USING FULL DATASET

Although there is some debate over terminology, we attempt a "replication" of Scaffidi and Chambers's work as defined by Bollen et al. [1] in that we use the original authors methods with new data. We deviate from Scaffidi and Chambers's methods in three ways: we include all Scratch projects instead of a sample; we include all projects from every user instead of sub-sample; and we include data on nearly every project, while Scaffidi and Chambers's analytic software fails to parse 18% due to errors and incompatibilities. In each sense, we include more data, increase internal validity, and work around limitations in the previous work. In all other ways, we attempt to ensure that our methods are identical to Scaffidi and Chambers's.

Our replication analysis began with the dataset of 1,925,054 public projects shared between March 2007 and April 2012. Following Scaffidi and Chambers, we first remove 506,072 projects that are remixes. Next, we remove 775,736 projects that were shared after July 1,

[2]Because a small proportion of users produce most projects, very active users will be overrepresented.

	Depth	Breadth
(Intercept)	74.72***	2.68***
	(0.95)	(0.00)
months	2.57***	0.01***
	(0.17)	(0.00)
R^2	0.00	0.00
Adj. R^2	0.00	0.00
Num. obs.	643246	643246

$^{***}p < 0.001$

Table 1. Regression models on measures of project depth and breadth. The unit of analysis is average month of user experience.

2010, which appears to be near the date that Scaffidi and Chambers collected their dataset.[3] Our final dataset includes 643,246 projects.

Following Scaffidi and Chambers's methodology, we calculated measures of depth, breadth, and months of creator "experience" for every non-remix project in Scratch from March 2007 to July 2010. Our linear models are presented in Table 1, where the regression results are opposite in sign to those reported by Scaffidi and Chambers. We also estimate t-tests between users' first and last months and find that, like Scaffidi and Chambers, our results are positive in sign. That said, because we use a much larger sample (23,092 users instead of 145), we are easily able to reject the null hypothesis of no relationship for both depth ($\Delta\mu = 39; t = 13.04$) and breadth ($\Delta\mu = 0.15; t = 8.24$).

Because these analyses involve multiple projects from the same users, whose projects' breadth and depth may be correlated over time, they violate regression's assumption of independent observations. Although this error affects Scaffidi and Chambers's original analysis, it is aggravated in our analysis which includes all projects. We address this important threat in our alternative analysis in the section following the next.

SIMULATION: REPLICATION USING SAMPLE

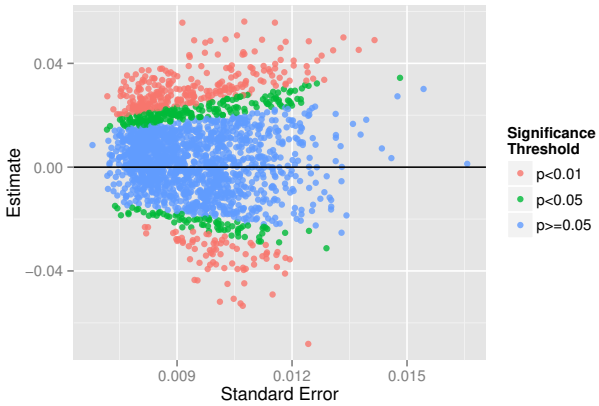

Figure 2. Plot of estimate vs. standard error from simulating samples using the Scaffidi and Chambers methodology.

The fact that our regression results are opposite in sign to the well estimated results reported by Scaffidi and Chambers is surprising. One explanation may be due to one of the other major findings in Scaffidi and Chambers paper: a high "dropout" rate in the Scratch community, where most Scratch users share only a very small number of projects. Even though the "Suprise Me" feature oversamples on very active users, Scratch's steep dropout rate means that only a small number of users contribute data for later months and that, as a result, the results are very sensitive to the particular users that are sampled. This is further aggravated by the small sample size which previous work has shown can lead to incorrect estimates [7]. To test this theory, we implemented the sampling technique used by Scaffidi and Chambers (e.g., sampling on random projects, choosing 250 unique users; taking the first project and then one randomly selected project from each subsequent "page" of 15 projects, etc). Because Scratch is open source software, we reviewed the Scratch code from the time of Scaffidi's paper to verify that our random project selection matches the "Surprise Me" functionality used in the original study. Using this methodology, we created 2,000 samples of projects from 250 users and estimated the linear regression models for each sample.

A visualization of the resulting estimates for the size of standard errors versus estimates for breadth across all 2,000 simulated samples is shown in Figure 2. We find that in 70% of our samples, there is no statistically significant relationship at the conventional level of $\alpha = 0.05$. That said, in 13.5% of the simulations, we estimate a positive association at the $\alpha = 0.01$ level. In 3.8% of the cases, we estimate a negative relationship at the same level. Results for depth are substantively similar but even less likely to result in estimates for which we can reject the null hypothesis of no relationship. In the models that show negative associations between user experience and both depth and breadth at below the $\alpha = 0.01$ level reported by Scaffidi and Chambers, our parameter estimates are very similar to those reported by them. Although it occurs in less than 4% of our simulated samples for breadth and less than 1% for depth, we were able to reproduce the earlier work's results. It seems likely that Scaffidi and Chambers's surprising results are due to one of these atypical samples.

ALTERNATIVE ANALYSIS: WITHIN-USER ESTIMATES

With a much larger sample, we can also fit types of models that would require more statistical power than was available to Scaffidi and Chambers and that might take into account additional threats to validity. In particular, we use fixed effects multilevel models [15] to estimate variation in depth and breadth associated with changes within users' experiences over time, addressing the issue of non-independent observations present in Scaffidi and Chambers's analysis and in both sets of analyses

[3]Sensitivity analyses showed similar results with or without remixes or projects created after the study by Scaffidi and Chambers.

	ln Depth	Breadth
ln Months Experience	0.02***	0.10***
	(0.00)	(0.00)
ln Comments Since Last Project	0.01***	0.01*
	(0.00)	(0.00)
R^2	0.0002	0.0011
Adj. R^2	0.0001	0.0008
Num. obs.	643246	643246

$^{***}p < 0.001$, $^{*}p < 0.05$

Table 2. Regression models for Scratch users' depth and breadth. All models use user-level fixed effects and reflect within-user estimates.

presented above. These models are equivalent to fitting dummy variables for every user and control for every variable—observed or unobserved—that has a consistent effect on the outcome across a user's projects. In the case of depth, we also log-transform Scaffidi and Chambers's depth measure to better meet parametric assumptions of the model. By adopting a regression framework instead of relying on t-tests, we can add additional controls. For example, we can control for the number of comments received since the last project—a variable shown to be an important predictor of learning in Scratch [5].

Results from fixed effects models estimated using the full population of non-remix projects are presented in Table 2. In our breadth model we estimate that a 1% increase in the amount of user experience is associated with a 0.02% difference in the number of blocks. In our breadth model, a 1% increase in the amount of the user's experience is associated with a 0.001 unit difference in the number of categories used in a project, holding all else constant. Similar to Scaffidi and Chambers's t-tests, these results suggest small, positive relationships between experience and measures of project depth and breadth. Goodness of fit statistics suggest that neither model describes much of the within-user variation in breadth or depth, suggesting the need for better predictors and better measures.

DISCUSSION

Taken together, our replication, simulation, and within-user model offer one explanation for the surprising contradiction between findings by Scaffidi and Chambers and other research on learning in Scratch. Although Scaffidi and Chambers's 2012 paper suggested that Scratch users tend to share decreasingly code-heavy projects over time and that users' projects trend toward using a less diverse range of code concepts, our replication of their findings in the full dataset of Scratch users from 2007 to 2010 suggest that the actual relationship is reversed.[4] Our simulation results suggest that Scaffidi and Chambers drew an unlucky random sample. Finally, using within-user models, we found consistently positive relationships between experience and breadth and depth.

One potential takeaway is the need for improved quantitative measures of learning in Scratch. For example, subsequent work has offered a category system for blocks similar to Scaffidi and Chambers's measure of breadth, where Scratch blocks correspond to computational thinking concepts (e.g. loops, operators, and events) [3]. Other work has analyzed when users adopt these individual concepts in their Scratch code as a way to measure learning [5]. Scaffidi and Chambers's measures each require continued performance and consider that a user is not progressing if their projects do not include more code and a wider variety of code over time. More recent work has used trajectory-based measures of the cumulative repertoire of programming concepts [17, 5] which we believe is a promising avenue for future research.

Although Scaffidi and Chambers's surprising results on depth and breadth seem to have been driven by a small dataset and an unlucky sample, their paper includes many other findings and detailed analyses that remain important for researchers of learning and programming in informal environments. Of particular importance is their description of a high dropout rate in the Scratch community. While high attrition is common to many online communities, it has serious implications for learning outcomes when, as Scaffidi and Chambers rightfully point out, very few users stick around over a significant period of time. Whatever the modeled estimate of learning over time may be, many (or most) users do not participate long enough to receive any benefits.

We hope that our work helps demonstrate the value of replication in human-computer interaction research. The majority of replications in HCI have been described as unplanned replications of previous findings [6]. In this paper, we explicitly set out to conduct a replication, and our findings are opposite to those in the original study.

For designers of informal learning communities such as Scratch, this positive evidence of skill progression represents good news, provides an answer to the puzzle of Scaffidi and Chambers's surprising results, and brings their study into harmony with other research. Of course, our revised estimates only reinforce Scaffidi and Chambers's parting argument calling for research to help decrease high levels of "attrition" among users of informal learning environments. When users' skills progress over time, it is even more important that designers understand how to support and increase engagement.

ACKNOWLEDGMENTS

We would like to thank Christopher Scaffidi, who offered graceful, valuable feedback on this paper. We would like to thank the Lifelong Kindergarten group at the MIT Media Lab for creating Scratch as well as the millions of Scratch users who create and participate on the Scratch website. We would also like to acknowledge Mitchel Resnick, Andrés Monroy Hernández, and our anonymous reviewers for their thoughtful feedback. Financial support for this work came from the National Science Foundation (grants DRL-1417663 and DRL-1417952).

[4]In results not reported here, we found that our pattern of results is similar using a complete dataset from 2007 to 2012.

REFERENCES

1. Kenneth Bollen, John T. Cacioppo, Robert M. Kaplan, Jon A. Krosnick, James L. Olds, and Heather Dean. 2015. *Social, Behavioral, and Economic Sciences Perspectives on Robust and Reliable Science*. Technical Report. National Science Foundation. http://www.nsf.gov/sbe/AC_Materials/SBE_Robust_and_Reliable_Research_Report.pdf
2. Karen Brennan. 2013. *Best of both worlds: Issues of structure and agency in computational creation, in and out of school*. Ph.D. Dissertation. Massachusetts Institute of Technology. http://dspace.mit.edu/handle/1721.1/79157
3. Karen Brennan and Mitchel Resnick. 2012. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*. http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf
4. Aniket Dahotre, Yan Zhang, and Christopher Scaffidi. 2010. A Qualitative Study of Animation Programming in the Wild. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '10)*. ACM, New York, NY, USA, 29:1–29:10. DOI: http://dx.doi.org/10.1145/1852786.1852825
5. Sayamindu Dasgupta, William Hale, Andrés Monroy-Hernández, and Benjamin Mako Hill. 2016 (forthcoming). Remixing as a Pathway to Computational Thinking. In *Proceedings of the 19th Conference on Computer Supported Cooperative Work & Social Computing (CSCW '16)*. ACM.
6. Kasper Hornbæk, Søren S. Sander, Javier Andrés Bargas-Avila, and Jakob Grue Simonsen. 2014. Is Once Enough?: On the Extent and Content of Replications in Human-computer Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3523–3532. DOI: http://dx.doi.org/10.1145/2556288.2557004
7. John P. A. Ioannidis. 2005. Why Most Published Research Findings Are False. *PLoS Med* 2, 8 (2005), e124. DOI:http://dx.doi.org/10.1371/journal.pmed.0020124
8. Jay Lemke, Robert Lecusay, Michael Cole, and Vera Michalchik. 2015. *Documenting and assessing learning in informal and media-rich environments*. MIT Press.
9. John H Maloney, Kylie Peppler, Yasmin Kafai, Mitchel Resnick, and Natalie Rusk. 2008. Programming by choice: urban youth learning programming with Scratch. *ACM SIGCSE Bulletin* 40, 1 (2008), 367–371. DOI: http://dx.doi.org/10.1145/1352135.1352260
10. Andrés Monroy-Hernández. 2007. ScratchR: sharing user-generated programmable media. In *Proceedings of the 6th international conference on Interaction design and children (IDC '07)*. ACM, New York, NY, USA, 167–168. DOI: http://dx.doi.org/10.1145/1297277.1297315
11. Kylie A. Peppler and Mark Warschauer. 2011. Uncovering Literacies, Disrupting Stereotypes: Examining the (Dis)Abilities of a Child Learning to Computer Program and Read. *International Journal of Learning and Media* 3, 3 (2011), 15–41. DOI: http://dx.doi.org/10.1162/IJLM_a_00073
12. Alexander Repenning, David C. Webb, Kyu Han Koh, Hilarie Nickerson, Susan B. Miller, Catharine Brand, Ian Her Many Horses, Ashok Basawapatna, Fred Gluck, Ryan Grover, Kris Gutierrez, and Nadia Repenning. 2015. Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools Through Game Design and Simulation Creation. *Trans. Comput. Educ.* 15, 2 (April 2015), 11:1–11:31. DOI: http://dx.doi.org/10.1145/2700517
13. Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67. DOI: http://dx.doi.org/10.1145/1592761.1592779
14. Christopher Scaffidi and Christopher Chambers. 2012. Skill Progression Demonstrated by Users in the Scratch Animation Environment. *International Journal of Human-Computer Interaction* 28, 6 (June 2012), 383–398. DOI:http://dx.doi.org/10.1080/10447318.2011.595621
15. Judith D. Singer and John B. Willett. 2003. *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence* (1 ed.). Oxford University Press, USA.
16. Max L. Wilson, Wendy Mackay, Ed Chi, Michael Bernstein, Dan Russell, and Harold Thimbleby. 2011. RepliCHI - CHI Should Be Replicating and Validating Results More: Discuss. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*. ACM, New York, NY, USA, 463–466. DOI: http://dx.doi.org/10.1145/1979742.1979491
17. Seungwon Yang, Carlotta Domeniconi, Matt Revelle, Mack Sweeney, Ben U. Gelman, Chris Beckley, and Aditya Johri. 2015. Uncovering Trajectories of Informal Learning in Large Online Communities Of Creators. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*. ACM, 131–140. DOI: http://dx.doi.org/10.1145/2724660.2724674