BootUp Professional Development Evaluation

2020-21 Full Report

July 2021

Evaluation by:

Peter Rich, PhD
Brigham Young University

1. Executive Summary

During the 2020-2021 school year, BootUp partnered with local education agencies (LEAs) to support BootUp's mission to prepare and empower teachers and implement elementary coding and computer science programs nationwide. BootUp is dedicated to empowering elementary teachers and students in underrepresented and underserved communities (see: bootuppd.org/about). BootUp pursues these goals by implementing equitable and sustainable computer science initiatives using a comprehensive approach. BootUp supports district administrators, coaches, and teachers through district-level support and district-wide professional development (PD) for elementary teachers on how to teach elementary computer science. By preparing teachers, BootUp and partner districts hope to create a sustainable and scalable model to bring computing to as many students as possible.

BootUp's PD consists of several workshops for teachers spread throughout the school year. Workshops are supported through teacher and student resources, hands-on activities, and in-classroom modeling. During the 2020-2021 school year, 21 school districts in 10 different states participated in BootUp's continuous PD. This report provides an independent analysis of BootUp's efforts to train teachers in the districts that completed a year of PD during the 2020-2021 school year.

There are 7 key findings from the efforts of BootUp with teachers during the 2020-2021 school year. Namely...

1. Nearly 50,000 elementary students across 10 states began to learn to code as a result of BootUp partnerships in 2020-2021.
2. Of those, demographics at participating schools suggest approximately 33,000 students eligible for Free or Reduced Lunch (66%) and over 40,000 minority students (82%) have begun to learn to code in 2020-2021. And that overall, approximately 45,000 students (91%) are students from under-represented groups in computer science education;
3. COVID conditions affected teachers' ability to implement coding lessons in the classroom. While over a third of teachers began teaching coding weekly, teaching occurred evenly across all elementary grades (K-5), indicating school-wide implementation;
4. Elementary teachers overwhelmingly reported that their greatest success was that their students loved to code;
5. Elementary teachers demonstrated significant growth in their confidence and competence to teach coding, showing the *greatest* confidence with sequences, loops, and algorithms;
6. Participating teachers were *least* confident with functions, abstractions, conditionals, variables and computational thinking practices; and

7. Participating teachers found the BootUp Professional Development to be highly effective, crediting its hands-on approach with increasing their confidence to code significantly and strongly.

1.1. BootUp Professional Development Model During COVID-19

BootUp works with school districts to provide continual support and professional development over the course of 3 years. BootUp's support consists of the following components:

(a) ongoing collaboration with district administration and personnel to design, study, and iterate on an equitable implementation;
(b) training and working with one, or several, district instructional coaches to support sustainability of the initiative;
(d) training teachers through either virtual or in-person professional development workshops in the summer and throughout the school year;
(e) providing open teaching resources to teach elementary coding (see https://bootuppd.org/curriculum); and
(f) on-site visits for demonstrations and co-teaching opportunities for teachers (when available and able due to travel permissions).

Ideally, this model is executed over a 3-year period in a scaffolded manner, described by year below:

Year 1: BootUp takes responsibility for the majority of training, including multi-day and monthly professional development (PD) meetings. The district instructional coach attends and is invited to participate in the planning and execution of these meetings, as well as participate in a virtual Instructional Coach Course; however, they are not typically expected to take charge and train the teachers in Year 1. The district instructional coach is a district employee who serves as the main contact between BootUp and their district's teachers who oversees the training of computing teachers.

Year 2: The district instructional coach begins to share the responsibility for co-teaching professional development meetings and supporting teachers in their classrooms, carrying a heavier load of this teaching as the school year progresses.

Year 3: The district coach assumes primary responsibility for delivering continuous professional development, with BootUp personnel playing a support role. This prepares district coaches to train new coding teachers as they come along, and to know how to support teachers in their learning. The intent is to provide a model that is both scalable and sustainable. By the end of the third year of professional development, BootUp hopes to have sufficiently developed the capacity within the district to train coding teachers without its help.

2. Methods

This evaluation was performed by Peter Rich, PhD, of Brigham Young University. Dr. Rich teaches undergraduate and graduate courses on the design, development, and psychology of instruction at the McKay School of Education. Dr. Rich's research over the past decade has focused on how to teach coding to elementary-aged children, resulting in several publications on this topic. For this evaluation, BootUp coordinated the desired outcomes and research questions with Dr. Rich. BootUp has allowed Dr. Rich complete independence to analyze and interpret data collected via teacher surveys. While Dr. Rich coordinated with BootUp to ensure data integrity, the analysis of the data presented consists of Dr. Rich's independent interpretation and recommendations based on that data.

2.1. Data collection

At the start of the first training of the year, teachers completed an instrument called the "Teachers' Beliefs about Coding and Computational Thinking" (TBaCCT). The TBaCCT is a validated instrument created to measure changes in teachers' beliefs about teaching coding and computational thinking in the classroom (Rich, Larsen, & Mason, *2020*). At the final BootUp workshop of the year, teachers again completed the TBaCCT. They also answered several additional questions about their: (a) coding-related teaching practices throughout the school year, (b) confidence with specific coding concepts, practices, and perspectives, (c) personal experiences teaching coding, and (d) feedback and evaluation of BootUp's professional development.

Finally, after analyzing pre and post data, Dr. Rich contacted teachers who agreed to further follow-up. These teachers completed an additional survey targeting their practices in integrating coding in the classroom.

2.2. Data Analysis

Both quantitative and qualitative data were collected via pre and post surveys for this evaluation. Quantitative data were primarily summarized using descriptive statistics, while qualitative data were analyzed by using an emergent cross-comparative analysis (Rich, 2012).

3. Results

BootUp worked with 185 total schools across 21 different districts in 10 states in 2022-21. Seven of these (Suburban_5, Suburban_3, Rural_2, Rural_6, Rural_1, Rural_3, and Rural_4) are currently already reaching 100% of their schools. Six of the remaining districts have designed implementation models that allow them to reach every school by their third year (Suberban_1, Urban_2, Urban_3, Mixed_1, Urban_5, Suburban_4). Table 3.0.1 reports on aggregated school-wide demographics collected from the NCES on schools that participated in 2020-21.

Table 3.0.1
NCES Demographic data for Participating Districts

District	Participating Schools	Total Students (entire school)	FRE + Direct Certification	FRE %	Title I or FRE >40% Schools	Minority Students	Minority Students %
Suburban_1	19	15,140	13,420	89%	19	14,766	98%
Rural_1	5	2,842	1,655	58%	5	1,188	42%
Rural_2	1	127	45	35%	1	13	10%
Urban_1	10	4,294	3,894	91%	10	4,253	99%
Urban_2	28	13,617	12,030	88%	27	12,837	94%
Urban_3	3	939	667	71%	2	668	71%
Suburban_School_1*	1	452	296	65%	1	451	100%
Rural_3	2	602	189	31%	1	126	21%
Rural_4	1	268	177	66%	1	268	100%
Mixed_1	23	11,546	5,206	45%	18	6,308	55%
Rural_5	1	55	24	44%	1	15	27%
Suburban_3	5	4,584	2,260	49%	5	2,570	56%
Urban_4	9	3,439	2,287	67%	8	2,794	81%
Urban_5	20	11,913	5,995	50%	15	9,788	82%
Suburban_4	10	6,191	3,904	63%	7	5,681	92%
Urban_6	36	21,625	12,551	58%	35	19,824	92%
Suburban_5	3	1,419	1,061	75%	3	1,252	88%
Suburban_6	4	2,506	775	31%	1	819	33%
Urban_School_1*	1	247	234	95%	1	241	98%
Urban_School_2*	1	392	375	96%	1	383	98%
Urban_School_3*	1	327	315	96%	1	322	98%
21	**184**	**102,525**	**67,360**	**66%**	**163**	**84,567**	**82%**

*Individual Schools that were treated as their own district.

Of the 184 schools impacted, 163 are either Title 1 eligible and/or have 40% or more students eligible for free and reduced meals (FRE). The percentage of FRE students or Direct Certification at the schools impacted is 66% overall. Consequently, this suggests approximately 33,000 students (66% of total students impacted) who have begun to learn to code this year due to the BootUp partnerships qualify for free and reduced meals. And that over 40,000 minority students (82% of total students impacted), including students who are Black, Hispanic, Asian, Pacific Islander, American Indian/Alaska Native, and of Two or more races, learned to code due to this program. In all, that suggests that approximately 45,000 students (91% of participating students come from traditionally under-represented student groups in computer science education.

While Table 3.0.1 demonstrates the entire population of participating districts, actual impact numbers from the 2020-21 school year that were reported by districts appear in Table 3.0.2. This provides a more accurate representation of the number of schools and students immediately impacted during the 2020-21 academic calendar. Over 49,000 students and 639 teachers were directly impacted by BootUp training in 2020-21 across urban, suburban, and rural districts. Due to the unique circumstances of 2020-21 brought on by COVID-19 restrictions, many locations split workshops into smaller sessions (e.g., 90 minutes after school). However, workshops were counted as complete when teachers participated in 3 hours of training. Furthermore, teachers in lower elementary (K-2) participated in different workshops than those in upper elementary (3-5) in order to delve deeper into grade-level specific materials.

Table 3.0.2
Impact Data as Reported by Participating Districts in 2020-21

		New Participation 2020-21			PD Workshops 2020-21	
District Name:	**State**	**Schools**	**Teachers**	**Students**	**K-2**	**3+**
Suburban_1	TX	19	66	8,519	3.0	3.0
Rural_1	MD	5	11	3,700	3.5	3.5
Rural_2	CO	1	8	118	1.0	4.0
Urban_1	IL	10	39	4,609	0.0	3.0
Urban_2	TX	29	58	7,768	3.0	3.0
Urban_3	AL	3	183	1,947	1.0	1.5
Rural_3	CO	1	4	589	3.0	3.0
Rural_4	WY	1	11	137	0.0	1.5
Mixed_1	TN	24	54	7,563	2.0	2.0
Rural_5	CO	1	10	43	3.0	3.0
Suburban_3	NY	5	7	0	1.0	1.0
Suburban_School_1*	GA	1	2	508	3.0	0.0
Urban_4	NY	9	11	486	3.0	3.0
Urban_5	NY	19	45	2,418	3.0	3.0
Suburban_4	MD	10	18	3,413	3.0	3.0
Urban_6	TN	37	49	4,924	0.0	4.0

Suburban_5	WA	3	3	0	0.0	3.0
Suburban_6	CO	4	30	1,800	3.0	3.0
Urban_School_1*	NY	1	12	170	5.0	5.0
Urban_School_2*	NY	1	7	300	5.0	5.0
Urban_School_3*	NY	1	11	205	3.5	3.0
Total:	**10**	**185**	**639**	**49,217**		

Note on discrepancies

The reader may note that there are a few discrepancies between the reported numbers in Tables 3.0.1 and 3.0.2. These are due to changing demographics in some schools after NCES data were reported. For example, in two different districts, a new school was being built and did not appear in the NCES data, but did end up participating in BootUp workshops. In another district, two schools merged into one between NCES data reporting and BootUp participation. The numbers represented are otherwise accurate according to NCES data and participation reported by each district.

3.1. Participating Teacher Demographics

Table 3.1 contains information for all school districts that completed at least 3 of BootUp's 8 planned PDs during the 2020-2021 academic year. In addition, these numbers represent teachers who completed the pre and post surveys, respectively. In many districts, not all teachers completed the post survey. Because not all teachers completed either one survey or the other, these numbers are lower than the numbers of teachers actually impacted by BootUp partnerships in 2020-2021. The remainder of the data in this report comes from these responses.

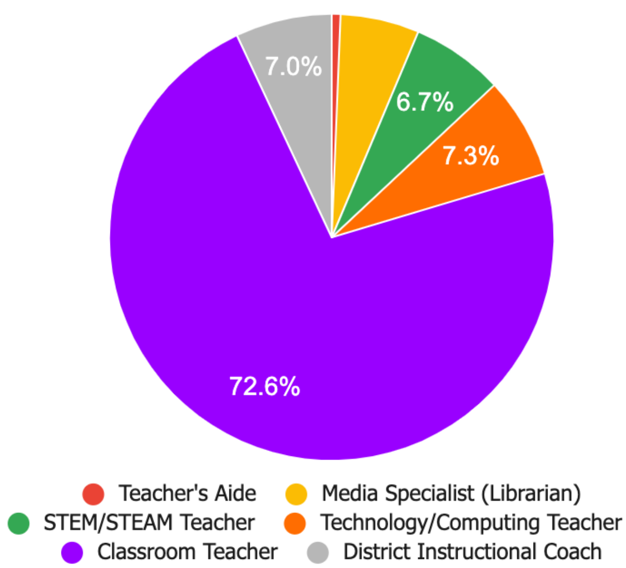
Table 3.1.
Demographics of BootUp teachers for 2020-2021 academic year

School District	State	# of schools that completed surveys (pre/post)	# of Teachers Trained (pre/post)	# of students taught*
Suburban_1	TX	9/8	32/8	8,519
Urban_1	IL	8/8	29/13	4,609
Urban_2	TX	33/12	47/13	7,768
Mixed_1	TN	13/14	34/25	7,563
Suburban_3	CO	1/1	2/5	43
Urban_4	NY	10/7	10/8	486
Urban_5	NY	26/19	33/26	2,168

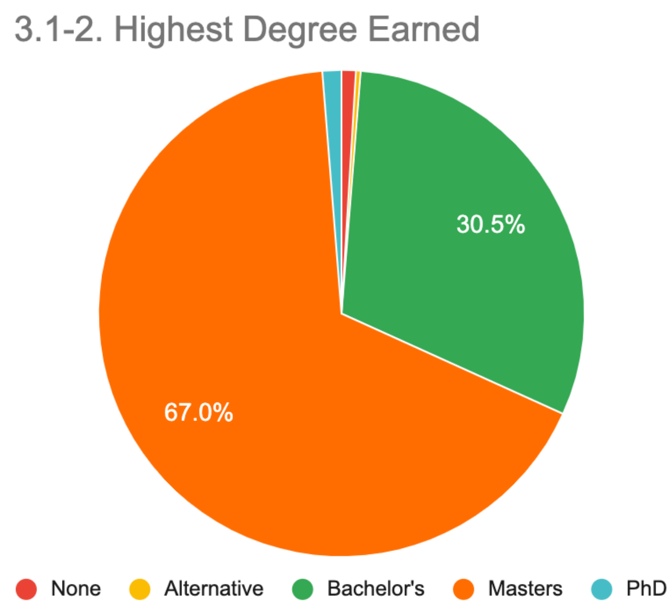
Suburban_4	MD	11/8	16/11	3,413
Urban_6	TN	21/21	28/24	4,924
Suburban_6	CO	5/6	21/25	1,800
Total		**137/104**	**310/164**	**41,293**

The vast majority of teachers were female (86.9%) and were classroom teachers (72.6%) (see Figure 3.1-1). Classroom teachers are certified teachers who are given the primary responsibility of educating students in each of the main subject areas in elementary school. In addition to classroom teachers, the remaining participants identified as a District Instructional Coach, Media Specialist (i.e., Librarian), STEM/STEAM specialist, or Technology/Computing specialist. Unlike classroom teachers (who are generalists), specialists teach 1-2 specific subjects and often see students from many different classes or age groups throughout the day. District Instructional Coaches are responsible for ensuring teachers are trained to teach coding and computational thinking in each district.

3.1-1. Professional Teaching Role



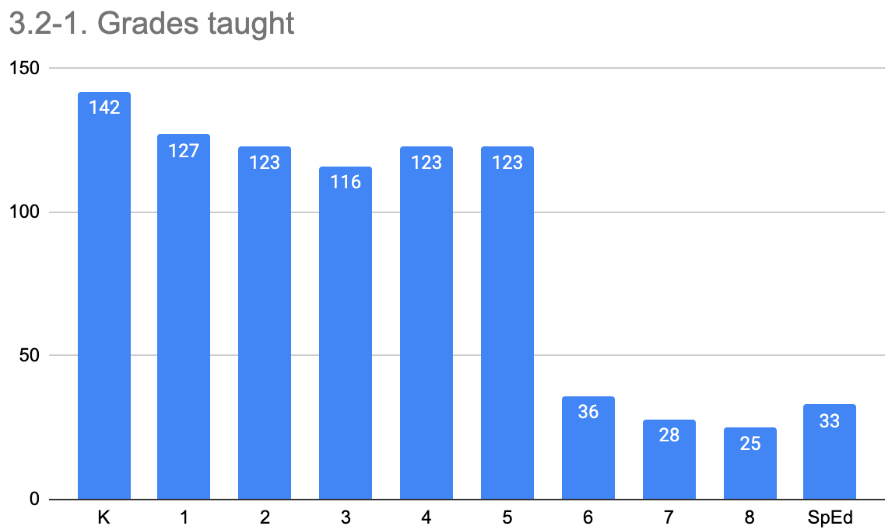
The majority of participants were well-educated (see Figure 3.1-2), with two-thirds having earned a Master's degree and roughly 30% having earned a Bachelor's degree. At the beginning of their participation with BootUp PD, teachers reported an average of 12.38 years of teaching experience, with 9.94 years experience teaching the same grade of students. While teachers' overall teaching experience could thus be categorized as veteran, their experience teaching coding stood in stark contrast at 2.2 years overall. Thus, while some teachers have been teaching coding prior to participating in BootUp training, their experience with coding appears to be relatively novice.



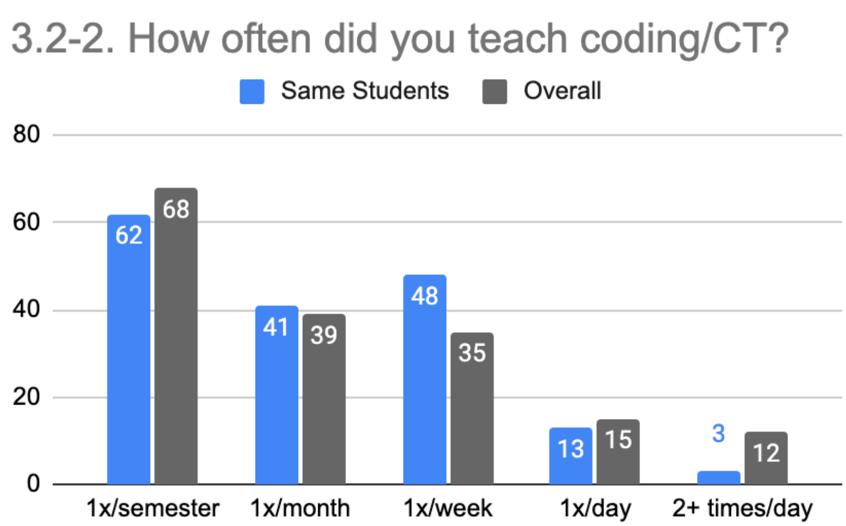
3.2. Teaching Practices

In this section, I report on teachers' actions taken in regards to teaching coding in the classroom. The data primarily come from the 134 responses received on the survey administered at the final BootUp PD of the year.

First, while the majority of teachers were classroom teachers, there was nearly equal representation across all elementary grades (typically, K-5), as well as a few specialists who also taught some children in higher grade levels or special education (see Figure 3.2.1). This suggests that there was an even district-wide implementation of coding across all elementary grades, which supports BootUp's goal of providing coding instruction for all students.

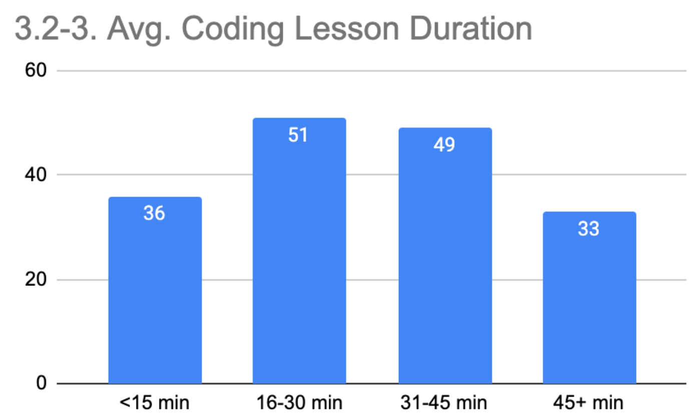


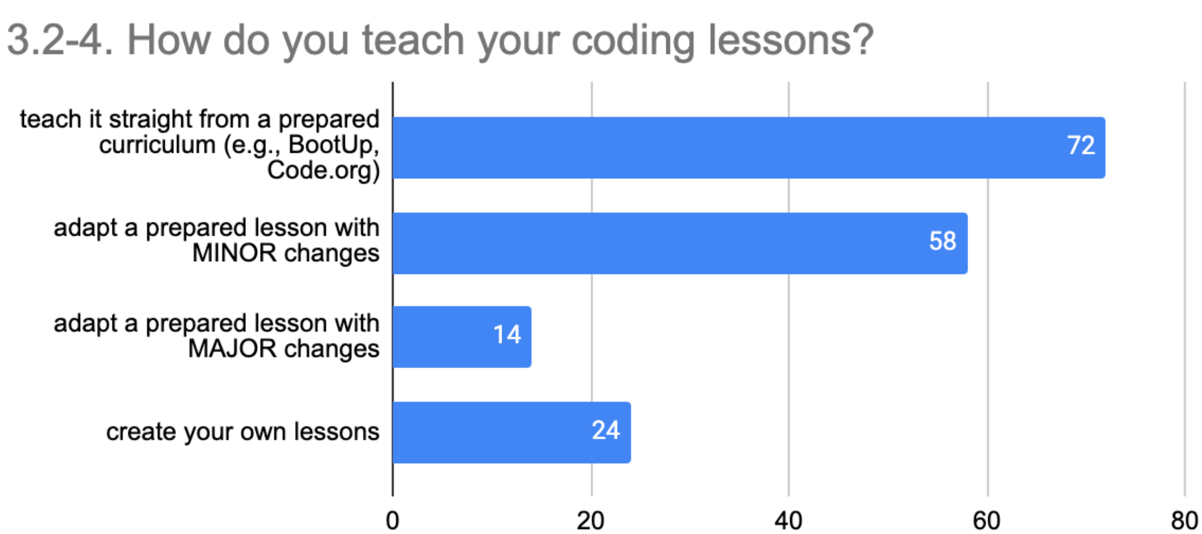
The 2020-2021 school year dealt with the unique challenges brought on by COVID-19 conditions. Under these conditions, 36.7% of teachers taught coding at least weekly (see Figure 3.2-2). While a little over a third of teachers (37.9%) taught the same group of students to code at least weekly, 40.2% were only able to teach coding to the same students once a semester. Given the restricted teaching hours during COVID conditions, it is possible that teachers would be able to teach the same students more frequently in a normal school year.



There was also variation in the length of coding lessons, with almost half (48.5%) of lessons lasting 30 minutes or longer. Lesson length is an important consideration when it comes to coding, as longer lessons allow time for children to both explore and debug their coding programs. However, it's also important to consider that the majority of lessons in younger grades, such as kindergarten, usually last less than 30 minutes (see Image 3.2-3), due to the fact that children are just learning to sit still through school. Thus, it's possible that many of the shorter lessons are occurring in the younger grades.

Inasmuch as coding is new to most teachers, it's helpful to understand how they prepare and teach their coding lessons. Figure 3.2-4 demonstrates that most (77.4%) teachers used existing curricula, either straight from the lesson or with only minor changes. This highlights an important point; namely, that novice teachers look to existing resources in order to start teaching what is otherwise a new subject for them. Thus, it is important to provide teachers with such resources in order for them to be successful.





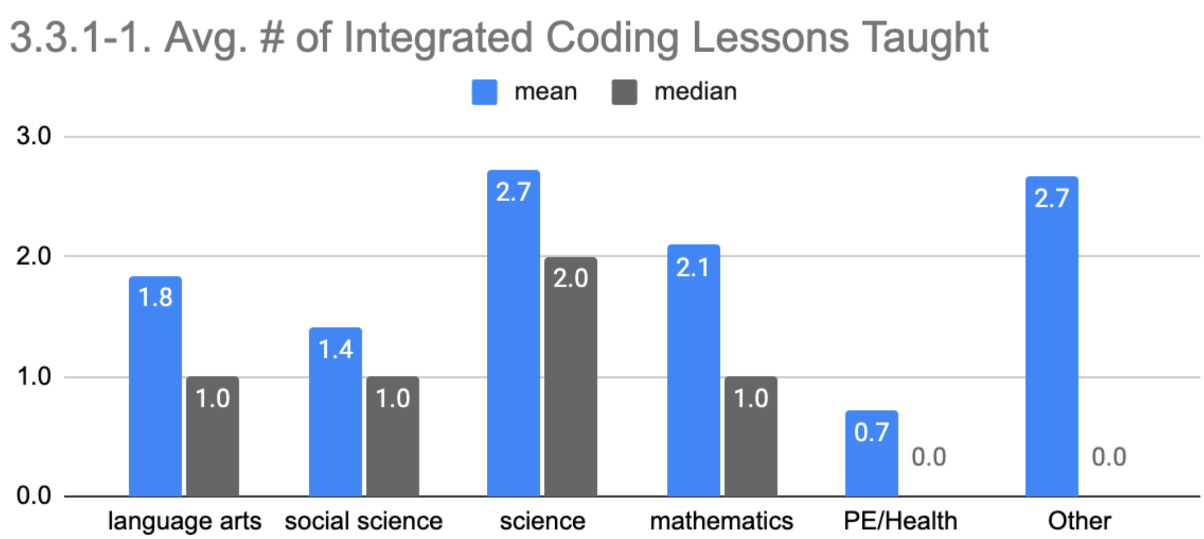
3.3. Classroom Coding Experience

Teaching coding can be challenging, as highlighted by teachers' comments about persistence. In addition to being a new subject for many teachers and a new way of thinking, learning to code often requires debugging. In this section, I first present teachers' comments about integrating coding into regular classroom instruction. I then discuss the successes and challenges that teachers faced.

3.3.1. Integrating Coding

On average (using the mean), teachers integrated coding into 12.7 lessons across the year (see Figure 3.3.1-1). However, I noted that there was a fairly substantial difference between the mean and the median due to a handful of fairly active teachers at integrating coding. The median is more likely representative of the average teacher overall. According to this data, the median number of integrated lessons was 5 lessons throughout the year, which is a good start with a new subject. Due to the fact that most of these are classroom teachers, they must find a way to teach coding while also teaching their regular core subjects. Thus, coding is likely a way to enhance those subjects rather than a topic unto itself. As one teacher put it, "I use the coding as presentation of material learned. Students will use coding to show the audience their knowledge of a topic."

The most common subject to integrate with was science, with math being the next most common. One teacher explained this by stating, "Coding is instructed more in the Science and Math class centered around a project involving engineering and measurement with distance and angles." Many teachers reported integrating coding in subjects other than language arts, social studies, science, mathematics, or PE/Health. This pattern of integrating lessons is promising, as it demonstrates that classroom teachers are likely to incorporate coding into their teaching in a variety of ways.



91 teachers provided information about their experience integrating coding into the classroom. Of these, 26% (n = 24) provided statements indicating that "students love it," as illustrated by the following comments:

- The students love it and they **catch on extremely fast**. They find so many things on their own without being taught. They take more risks.
- **Students adapt well in integrating coding** with other subjects
- The experience has been good, **students really enjoy coding** and creating projects.
- The students enjoy it. Connections can be made to what they are learning in content areas, what they are interested in, and coding. **Students are seeing how coding is a tool that helps them make things that interest them**.
- I was struggling with teaching coding because I wanted to master the skill before sharing with my students. Coding for me was difficult. I waited to share the Scratch Link because I wanted to master coding. However, **once I shared the link with the classes the students were thrilled [and had] fun with something new!**
- My experience with **integrating coding has been awesome!** The students have another way to differentiate their learning experience.

- The **students are more engaged** and showed improved mastery of the lessons.
- The students really enjoy it, some more than others. It **really helps to build a solid foundation of problem solving strategies** to use when approaching different coding platforms/projects.
- The students absolutely love coding and I think it is a great way to have student led discussions which **empowers them to be more independent!**
- It has been both **interesting, challenging, and fun.** Students have really enjoyed Scratch and the fact that some of them are virtual makes things that much more interesting.
- I use **coding and CT lessons mainly with mathematics.** The students seem the most engaged with it in this subject.

3.3.2. Main Successes

When asked what successes they have had in teaching coding and computational thinking, teachers overwhelmingly focused on students. Of the 142 answers received, 72% (102) emphasized increased student interest (53), knowledge (19), engagement (16), or success (14). The following quotes demonstrate these responses in each of these top four categories.

3.3.2.1. Student interest

I defined student interest as comments that refer to how interested students were in coding, their excitement, or how much fun they had with it. The following comments demonstrate some of the statements teachers made about student interest.

- **Students are vested and highly interested** in working on Scratch, which aids in motivation and attention span.
- Students developing **a genuine interest and passion for computer science**.
- Seeing **students really enjoy it, despite their success/challenges** with it.
- During a non-Covid year, it's a popular topic. I feel like **it helps me recruit students to take my tech electives**.
- **Students love coding and learning about how technology works**. Students learn not only how to code but they learn how to problem solve. They learn how to look at problems in different ways to determine how to solve them.
- **Students LOVE to code and look forward to our work.** I am able to have students at different abilities work at their level successfully. Through pairing and the challenge given to them they can be successful.
- The **kids enjoy coding and they work well together when coding**. We recently completed an interactive collage related to our opinion unit.
- My **students really enjoy coding**. They are more advanced than I previously thought them to be. They enjoy being creative and hands-on.
- I used a coding lesson in my formal observation. The **students got so involved that even my assistant principal joined** and do the coding activities.
- **Students are excited** when they are allowed to be creative, especially since they have learned so much about technology due to remote learning.
- My **first and second graders love coding and want to continue it.** During the Christmas season my students did coding using the Google Santa Tracker website.
- A lot of success. The more I do, the better they get. **I notice some students who enjoy coding are ones that I lose during other activities**.

3.3.2.2. Student Knowledge

I defined student knowledge as a notable increase in students' skills, ability to code or cognitive function. This ranged from basic computing concepts (algorithms, loops, logic) to soft-skills (communication) in kindergarteners to middle-schoolers. Some teachers also mentioned integration with other subjects. In all, teachers seemed to be impressed with what knowledge their students picked up and retained.

- Children have been able to create an **algorithm**
- students learning **logical operators**
- Students have gotten opportunities to present their work which in turn helps improve their **communication skills.**
- Students are **able to solve a math problem and code to the answer.**
- Students are successful with understanding the **basic concepts of coding.**
- Students **finally 'get' what coding is** all about.
- **Understanding of blocks**
- I have observed my students have all kinds of reactions to programming computer science curriculum. I believe my major success is **teaching my Kinder class how to program in Scratch.**
- My **students remembered what they had previously learned** and were able to quickly create their own projects.
- My students have created interesting, **content focused projects that demonstrate their knowledge in math & science**.
- The students have **created several great projects and are growing every time** we use this program.
- Students have g**ained comprehensive understandings of basic CS concepts** in grades ranging from pre-k to 6th grade. They have been understood concepts in other content areas through the integration of CS into their curriculum.
- My **kindergartners have been incredible in picking up the concepts and taking risks!**
- My s**tudents have great background knowledge in coding** because they do a lot of it in STEM class.
- Students are **quick to understand coding**.
- Had some **success in building knowledge**, primarily by using block coding with our elementary students, develop some **basic vocabulary and increasing problem-solving skills.**

3.3.2.3. Student engagement

Student engagement is similar to student interest. It is distinguished not by an expression of interest, but rather by a reference to students' behavior in class being one of focused activity.

- I learned about the different blocks, my students were **amazed about the level of engagement and collaboration**. The students took what they learned and they were able to become student leaders to help others.
- The **student engagement is always a success.** Students sometimes get frustrated but they are there for one another and help each other.
- Students **were really engaged**
- Student **engagement**.
- The **students are engaged and excited.** They love learning hands on.
- **Student engagement** and excitement.
- **Kids engaged.** Kids follow directions
- **Engaging students**
- Some students have **learned to be more focused and asked more questions. Other students shared more than they usually would.**
- Students **became more engaged in lessons.** Student behavior improved because they wanted to participate. Was able to create a coding club. Have the tools needed to teach kinder students to code and integrate skills into lessons.

3.3.2.4. Student Success

Student success refers to instances in which teachers mentioned students accomplishments related to coding, as exemplified by the following statements.

- Some **students exceed expectations!**
- One of my students created a project and **won the Computer Design contest** in the district.
- Watching the **students make the connections** with what happens when you connect codes or make something do something
- The students have **strengthened their problem solving skills and their ability to adapt their thinking** in different ways if something is not working correctly.
- I have been coding with Kindergarten. **It is incredible how quickly they caught on and retain the information** for the next project.
- The **students have taken off and come up with projects of their own** when I have given very limited instruction.
- Students are successful with **understanding the basic concepts** of coding.
- There are endless things to learn and so, **students begin to teach each other and me about how to do specific things they've figured out on their own.**
- **All my students have experienced some successes** with the coding lessons.
- **Students are so proud of their projects**. They are overwhelmed that they are able to create their own game.
- Some **students have really taken off with the coding, especially one of my second grade girls,** and that has been fun to see.

3.3.3. Main Challenges

The challenges that teachers had teaching coding and CT were more varied than their successes. The top 5 challenges teachers experienced with teaching coding were: time (30), virtual learning (26), resources (24), teacher knowledge (22) and student knowledge (14). While time and resources are often mentioned as challenges for teachers implementing new topics (Rich, 2012), virtual learning was a challenge unique to the 2020-2021 school year that affected many teachers. The following excerpts provide some insight into how teachers talked about their top five challenges.

3.3.3.1. Time

Challenges related to time nearly always mentioned a lack of time to teach coding in the classroom.

- I have to **take the time and explore** how to do it myself first. **We don't have enough time** in the classroom to dedicate to Scratch. It takes time to develop a project.
- **finding the time to teach it with my special needs students** with everything else that has to be taught
- The vocabulary and **the TIME to teach.** It has also been a challenge because of Covid19. They need the opportunity to work closely with their peers.
- **The time** and incorporating coding with ease now that **I am not seeing students weekly in the library,**
- there's often **so many other things we have to get to** and we can't spend as much time as we'd like.
- **Finding the time to integrate it** into the schedule.

- I may **see students every day for an hour for just two weeks**, or for one hour each week. So, depending on the schedule, I don't always have the momentum I want to have with them when we start. Students need to first understand that their mindset needs to change...failure is just another try in coding and in STEM class in general. Once they understand that, there's no stopping them.
- For me it's time--I want to teach coding all year, but I have to follow the TN Digital Readiness curriculum guide. Because **I only see my classes every other week** (due to large number of classes), I don't see them very many times per year, so it's hard to do more than 4 or 5 units each year.
- **Not enough time for it.**
- Being able to teach when **I only saw the students once every other week. They forget what we did or where we left off.**

3.3.3.2. Virtual Learning

Many teachers mentioned that it was difficult to teach coding due to the unique challenges of having to teach virtually. In some cases, virtual learning was hampered by resource issues, while in other cases, it hampered teachers' pedagogical practices.

- Trying to get **both my virtual and in class students to code at the same time.**
- Teaching it virtually. **It is hard to help a student when you can't see where they are stuck.**
- Sometimes it's difficult for students to understand to debug and **it's a long process to talk it through with them in a remote setting**.
- **Having students in split locations as well as lag in presenting material** on shared screens created difficulty. Some practice in typing simple code like HTML was challenging when using tablets in lower grades.
- **Teaching coding in a virtual environment** has been quite the challenge.
- **Being online and trying to help students with sharing projects and screens.** That was a bit difficult and frustrating.
- **Virtually, it has been challenging to explain where things are on the screen.**
- **Teaching virtually and presenting to younger learner,** sometimes I had to move slowly. Further, sometimes scratch had glitches or worked slowly.
- The biggest challenge has been **the virtual aspect of it. The wait time to help is longer because it can take a lot of time to share screens to help.**
- In hybrid learning, **sometimes the students at home have poor connectivity and they got kicked out of Zoom from time to time.**

3.3.3.3. Resources

Challenges with resources were often with internet access or outdated/failing technology.

- Lack of **accessible technology**
- Our **firewalls** blocked me being able to use Boot Up videos. This challenge has been overcome.
- Having **enough laptops/ ipads** for the class
- Students not 1:1 with devices, which has changed. **Not enough robotic materials.**
- With this year being remote **many students did not have a device to work on coding apps or software**.
- My challenges have be just the **internet issues,** which are out of my control, the students have learned patience and to adaptability skills.
- Our **Scratch does not always work** so it is hard to get a lesson done
- The **computer devices staying in working conditions** so all the students can access the programs..

- **Resources**, time, getting others to integrate CS and CT into their instruction. Good ideas that I can borrow and alter.
- Students are having a hard time working on **outdated Chromebooks.**

3.3.3.4. Teacher Knowledge

Many teachers mentioned their own lack of coding knowledge as a challenge that they had to deal with. In many cases, teachers also mentioned that they were overcoming this challenge due to participation in BootUp PDs.

- I'm not a tech person, so **when the students ask me questions, I could not answer** some of them; however, I would let the students share with each other and help each other.
- I myself have never been tech savvy. **Learning all of the different algorithms and codes has been challenging** but I continue to debug and learn alongside (and from) my students!
- **I am not an "expert"** so my experience is limited.
- **Answering the questions**. Sometimes there are soooo many I can't keep up.
- I am a beginner so **I need to learn more too.**
- Some challenges have included the **students needing help that is beyond my own knowledge level** and issues with our district's internet and technology capabilities being limited.
- **I wanted to be an expert before I taught the information.**
- The fact that they are kindergartners and that **I am new to coding** sometimes creates a bit of chaos.
- I'm just learning as well so **I can't always answer their questions.**
- **Problem solving while I'm still learning-**but it's helping me learn faster!

3.3.3.5. Student Knowledge

Some teachers mentioned that students' knowledge of coding could be a challenge. Curiously, this challenge had two dimensions to it; sometimes it was students' lack of coding knowledge that was a challenge, while other times the fact that there were children with greater coding knowledge than other students or than the teacher herself presented a challenge to the teacher.

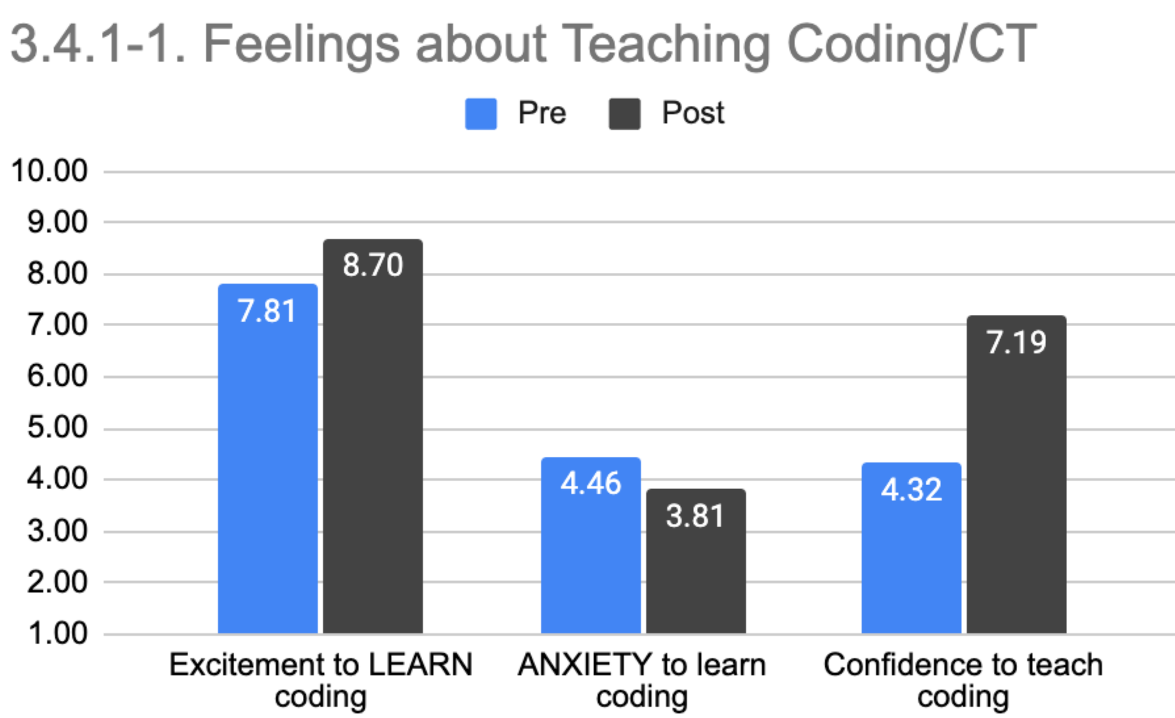
- **Wide range of knowledge** - some kids had already done Scratch, some were brand new. **Needed to help challenge both ends of the spectrum**. Those with more experience did help those who hadn't done it so that was good.
- The **student's ability to code has surpassed mine,** so I am not always helpful.
- The **students who "get it" will fly through challenges** to be ready to go on to the next level. This can be discouraging for some who are working.
- The kids are catching on super quickly and are **moving through the project very fast**
- Helping **students who know more than what I can assist them**
- Kids **not understanding how to do the broadcasting** part. Bringing sprites from other websites.
- Too many kids that don't understand and **asking questions at the same time.** A challenge I had was limited knowledge on my part to answer some of the kids' questions.
- I would love to have an opportunity to work with scratch jr. **There were moments Scratch was over my students' level just because they are 1st grade and many do not speak English.** That was probably my biggest challenge.
- Sometimes **it's difficult for students to understand to debug** and it's a long process to talk it through with them in a remote setting.
- It has been a challenge **teaching them to broadcast** and and how to pick a character to upload to their scratch program.

3.4. Confidence with Coding

In this section, I present changes in teachers' confidence and competence to teach computing, beginning with the TBaCCT (which was administered in the pre and post surveys), as well as a report on teachers' feelings about teaching coding and how these changed. These quantitative results are supported by teachers' qualitative answers to open-ended questions about the changes they experienced in their confidence to teach coding and computational thinking.

3.4.1. Teachers' Feelings about Teaching Coding

On the pre-survey, teachers answered questions about their excitement, anxiety and confidence to teach coding on a 10-point scale. They answered these same questions again after the BootUp training. Overall, teachers began with a medium-high level of excitement and increased this even more as they learned to code (see Figure 3.4.1-1).

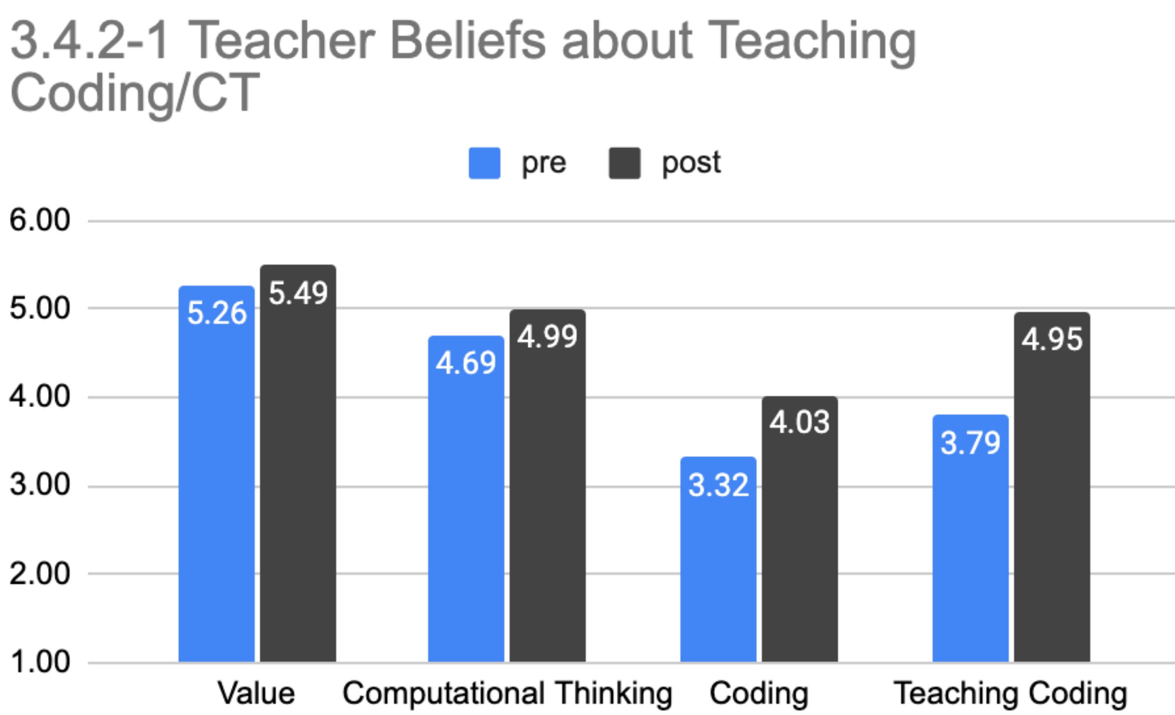


Perhaps just as important, teachers began with a medium level of anxiety about teaching coding, which saw a decrease by the last BootUp PD. The greatest change in teachers' feelings about teaching coding occurred in their self-reported confidence, which went from a lack of confidence (4.32/10) and jumped nearly 3 points to a 7.19/10. A t-test comparing pre and post scores demonstrated that changes in excitement ($p < .000$), anxiety ($p < .007$) and confidence ($p < .000$) were all statistically significant. Furthermore, thes changes were practically significant, though to differing levels. The was only a small effect on anxiety

(Hedge's g = -0.254), medium effect on excitement (Hedge's g = 0.444) and a strong effect on confidence (Hedge's g = 1.129).

3.4.2. Teachers' Beliefs

Teachers' beliefs about teaching coding were assessed using the TBaCCT, a 33-item scale that measures teachers' confidence in four areas: values, coding self-efficacy, CT self-efficacy and self-efficacy for teaching coding (see Figure 3.4.2-1). The TBaCCT uses a 6-point Likert scale with answers ranging from strongly disagree to strongly agree. As with teachers' feelings about teaching coding, teachers experienced statistically significant changes in each of these four areas ($p < .000$ in all measures)..



As demonstrated in Figure 3.4.2-1 teachers started with a high valuation of the importance of teaching coding. They felt that all students can and should learn to code and that learning to code will be important and useful for students' future careers. Teachers also showed higher, though less strong, confidence in their own ability to think computationally, falling between somewhat agree and agree on the 6-point scale in regards to questions such as, "I am good at solving puzzles," and "I can break a complex problem down into smaller problems." Despite starting higher, teachers still increased statistically significantly in each of these areas, though their effect sizes (which measures practical significance) were weaker (Hedge's g = .358 and .367, respectively).

In contrast, even though teachers began with a fairly high valuation of coding, their confidence to code and to teach coding started on the negative side. Perhaps due to teachers' 10+ years of teaching experience, they showed greater confidence with teaching coding, but did not quite reach the "somewhat agree" minimum confidence level in regards to statements about confidence to teach coding. And they disagreed with statements about coding more often than agreed to begin with. By the end of the first year, both of these categories increased, with teachers moving from the "somewhat disagree" to "somewhat agree" range in terms of coding questions, which showed a medium effect (Hedge's g = .580). While this is certainly moving in the right direction, there is still plenty of room left for teachers to grow in their confidence to teach coding. Ideally, teachers should move from a 4 to a 5 on this scale by the end of their second year.

Where teachers grew the most was in their confidence to *teach* coding. These questions were characterized by statements such as, "I know where to find the resources to help students learn to code," and "I can help students debug their computer programs." Teachers increased their confidence from less than "somewhat agree" to a solid "agree" on these statements, showing a significant ($p < .000$) and strong (Hedge's g = 1.063) practical effect. This is great progress and will likely enable teachers to grow their own confidence to code.

When asked, "how has your confidence to teach computing changed over the course of this year," 94% (124/132) of teachers indicated that it had increased. Teachers attributed this growth to three key things: hands-on practice, coding resources, and patient BootUp facilitators. The few teachers who did not indicate an increase mentioned that they were already comfortable with coding but that this training had provided them with additional, useful resources or ideas about how to integrate coding into the classroom.

To better understand teachers' changes in confidence to teach coding, I also asked them to respond to the statement, "BEFORE BootUp trainings this year, my feelings about coding were…" They then responded to the same statement for AFTER the trainings. I classified teachers' answers to identify patterns of how they felt about coding before and after the workshops. Table 3.4.2.a provides a side by side comparison of this comparison.

Table 3.4.2.a
Teachers' Feelings About Coding Before and After BootUp PD

Before PD	Frequency	After PD	Frequency
apprehensive	61	excited	47
ignorant	28	confident	44
positive	24	informed	20
dubious	9	integration	16
confident	7	comfortable	15
interested	6	positive	9
curious	5	wish there were more time	5
excited	4	interested	3
neutral	3	like it	1
overwhelmed	2	surprised	1
uninterested	2		
confused	1		
No integration	1		
Not confident	1		

The contrast between the before and after is clear. Initially, the majority of teachers were apprehensive about or ignorant of coding, whereas afterward, they were excited and confident. To be clear, there also several (24) teachers who initially indicated they were positive about coding. In many cases, even these teachers increased their excitement for coding. A selection of teachers' side-by-side comparisons of their before and after statements is provided in Table 3.4.2.b Each row represents the same person. Thus, the statement on the left was made by the same person as the statement to its right.

Table 3.4.2.b
Teachers' Statements about Coding Before and After BootUp PD

Before	After
I did not want to do it! I was scared to code.	I like to code now and use this as part of my lessons.
Positive	Even more positive now.
less involved into integration with other subjects	Ready for implementation and integration
I'm going to do this but I have no clue what I am doing,	I can do this with a little bit of patience and understanding of the program

I was interested in teaching students to code but did not have much background knowledge, vocabulary or experience needed to effectively do it.	I feel very informed and ready to teach students and other staff members about coding using Scratch Jr.
I was leary, because I did not have anyone to ask questions or see if I was doing the right things or using the correct terms.	I have grown so much and I have been able to expose my staff and students to coding across the content areas. It has become an incentive for many kids to finish their assignments to earn Scratch. I am proud of all that I have learned and I look forward to sharing with my school and other colleagues at any time.
I didn't know much about it	It is a highly engaging activity that can be used in many content areas
Extremely hesitant/not for me	Less hesitant/ willing to approach
positive	positive and more targeted about how we can integrate and teach specific concepts more confidently
scary and challenging	I feel determined
I can never do this!	I can do this!
Overwhelmed.	Much more comfortable.
I did not know anything about technology.	The kids will love this!
Neutral. I wanted to do it but didn't want to do it for the sake of doing coding. I wanted it to be purposeful.	I have learned how I can make coding purposeful in my classroom and activities that will benefit my students.
It was a little pointless to be honest.	Now I realize how coding is a way of developing your brain to work a different way.

3.4.3. Concepts, Practices and Perspectives

In addition to measuring changes in teachers' confidence to teach coding, I used another scale on the post assessment following Brennan & Resnick's well-known framework for assessing computational thinking (2012). This framework has been used by the Computer Science Teachers Association, the International Society for Technology in Education and the CSK-12 standards for teaching coding in K-12. I was unable to measure these at the beginning of teachers' training because they involve jargon-specific questions that teachers could only answer once they had learned about coding. Nonetheless, it is useful to know where teachers fall in their understanding of and confidence in each of these areas related to computational thinking. It will be useful to measure teachers' progress in these areas again at the end of their second year of training.

I used the same 6-point Likert scale for these questions (from strongly disagree to strongly agree). The first consideration is teachers' understanding of specific coding concepts. While teachers did not reach the "agree" threshold on any concept, all concepts nearly fell between 4 and 5 on a 6-point Likert scale. Teachers were most confident with sequences, loops, and algorithms, which makes sense since functions, variables and conditionals are not introduced until later lessons. It will be interesting to see how teachers' confidence with different coding concepts changes by the end of one more year of coding.

Table 3.4.3.
Teachers' Confidence for CT, Coding Concepts, Practices and Perspectives at year's end

Coding Concepts		CT Knowledge		CT Practices		CT Perspectives	
sequence	4.68	Pattern Recognition	4.49	collaborate around computing	4.36	Collaborating	4.99
loops	4.46	Algorithms	4.14	communicate about computing	4.32	Persistence	4.80
algorithms	4.24	Decomposition	4.04	foster an inclusive computer culture	4.22	Creating	4.74
functions	4.08	Evaluation /Analysis	4.01	recognize and define computational problems	3.86	Tinkering /Remixing	4.55
variables	4.04	Abstraction	3.78	create computational artifacts	3.57	Debugging	4.19
conditionals	3.92			test and refine computational artifacts	3.57		
				develop and use abstractions	3.51		

Teachers' knowledge of specific CT practices fell similarly mostly between 4 (somewhat agree) and 5 (agree). The one CT concept that teachers were less confident with was abstraction. Other research has demonstrated that this is a more difficult concept for teachers to grasp (Rich et al., 2021). Thus, they need continued practice with abstraction before I would expect to see more growth in this area.

CT *practices* are activities that one engages in when solving problems computationally. This was the CT-specific area where teachers were less confident, with more than half of their responses falling below the 4-point threshold. Once again, we see that teachers feel less confident in their ability to use abstractions, with this falling at the bottom of their list. Perhaps just as important to note is that teachers are nearly as unsure about their ability to create, test and refine computational artifacts. This is an important practice to promote as it involves the actual creation of code-based artifacts.

Finally, computational *perspectives* concern the way that developers perceive the world around them as they think computationally. In contrast to practices, this is the CT area where teachers' felt most confident, with all answers in the 4-5 range. Teachers were most confident in their ability to collaborate. Collaboration is a well-promoted 21st-century skill that has been promoted through other initiatives. However, "persistence" is a close second. Research (Rich et al., 2019) has demonstrated that teachers report higher levels of persistence or "resilience" among students who learn to code. This finding mirrors that, but at the teacher level. It is important that teachers learn to be resilient as they learn to teach coding and that they promote it amongst their students.

3.5. Teacher Evaluation of BootUp PD

As an organization, BootUp is always interested in improving its practices and adapting these to individual district needs. Consequently, I asked teachers to rate the different components of BootUp's PD and to compare it to other PDs that they had experienced. The big take-away from this part of the evaluation is that teachers were overwhelmingly pleased with the BootUp workshops. This section looks at their feedback for BootUp in greater detail.

On average, teachers attended 4.00 BootUp workshops this year. Most teachers completed three full BootUp PD workshops worth of content, which accounts for 9 total hours. One district (Shelby, TN) completed 12 hours, while another (Hamilton) completed about 6 hours of training for both K-2 and 3+ curricula.

Table 3.5-a breaks down each of the different components of the BootUp training and how teachers rated their experience with these on a 10-point scale, with 1 being the lowest and 10 being the highest.

During the pandemic, BootUp offered virtual model teaching, practice sessions, lesson planning sessions, and office hours in place of on-site support.

Table 3.5-a
Teachers' Ratings of the Different BootUp Components

BootUp Component	Definition	Rating
Hands-On Learning	Practice completing projects during PD	9.23
Videos	Walk-throughs of completed projects and how to work out each section of a coding project.	8.77
Coder Resource	Student materials, including video walk-throughs and debugging slides	8.75
Model Teaching	BootUp Facilitators visit teachers in their classroom and demonstrate teaching a lesson to students.	8.75
Sharing Projects/Ideas	Time given during PD to discuss and share ideas with peers.	8.61
On-site Coaching	BootUp facilitators visit teacher at their school and offer formative feedback/support	8.41
Discussion Forums	An online discussion forum for teachers to discuss specific coding questions and ideas.	7.88

When asked, "how does BootUp PD compare to other professional development experiences you've had," teachers rated it at 8.60/10, meaning BootUp compared favorably overall. I asked teachers to explain their overall BootUp rating, categorizing each response by using a 1-3 word description. One-hundred and five teachers provided a written rationale for their ratings. While there were a few people who indicated that they had difficulty with the online format, these were in the minority; the overwhelming response was positive. What's more, teachers' responses aligned with their ratings of the different BootUp components. Table 3.5-b reflects the top responses and the number of times they occurred.

The fact that the workshops were hands-on was the most consistent reason for teachers rating the BootUp workshops highly. Teachers also had high praise for their facilitators. In fact, there were no negative comments about any of the facilitators as a response to any question on any survey. Many teachers also pointed out that the PD was enjoyable, fun, or engaging. Likewise, about 10% of teachers highlighted the fact that the BootUp PDs provided them with the resources they needed to teach these lessons in the classroom. Another 10% appreciated that they could share their projects and ideas with other teachers, emphasizing the collaborative aspects of the workshops.

Table 3.5-b
Teachers' Rationale for Their BootUp Ratings

Description	Frequency	Example Quote
hands-on	38	The hands-on of the projects helps me see what problems my students might have and what to pre-teach
good facilitators	27	The facilitators...were the very best PD facilitators I've ever had. Their attentiveness, caring, and thorough presentations were not only encouraging, but inspiring. They are models of how teachers can and should be with students in general. And they really made learning coding a fun experience that I can imagine my students enjoying as well as I did. I want to be like them for my students.
engaging	23	It's hands-on and extremely engaging. This PD forces you to be the learner.
resources	14	The BootUp PD is easy to follow and they provide various resources that are helpful.
collaborative	13	The program is very collegial with the group working together to share ideas and help with debugging each other's work. There is sharing of resources for our classrooms and websites to enrich the student experience
learned something new	13	This PD was different from ones I attended in the past for the simple fact that I actually learned something new as opposed to being told to listen to a lecture in which the facilitator is paraphrasing a theory/strategy I already know about. The workshops here offered direct insight and suggestions for classroom implementation. The facilitator also did not approach me with a "I know more than you/I have more classroom experience than you" attitude which I have encountered in the past and is incredibly off-putting.

4. Conclusion

During the 2020-2021 school year BootUp teamed up with 21 school districts. Despite varying conditions and expectations brought on by the COVID-19 pandemic, teachers in these districts were able to complete 3 of BootUp's 8 PD workshops completely. Roughly 37% of these teachers began teaching coding at least once a week. Consequently, over 49,000 new elementary students were introduced to coding with nearly half of these receiving coding instruction once a week or more often. Approximately 91% of students at these schools are from groups underrepresented in computer science. While the vast majority of these teachers agreed on the importance of teaching coding to begin with, they also lacked confidence in their ability to code or teach coding. Through BootUp's hands-on training, teachers increased their confidence to teach code by 67%, showing strong and significant growth. Teachers described moving from a state of apprehension about coding to a state of excitement about teaching coding.

Despite these changes, there is still room for teachers to grow in their confidence to teach coding. They feel comfortable teaching sequences, loops and algorithms, but are still unsure about their ability to teach functions, conditionals, variables, abstraction and to promote computational thinking practices such as collaboration and debugging. While this report primarily represents the effect that PD had on -participating teachers, the effect it had on its students is best summed up by a statement that one of the teachers made: "the more I do, the better they get." As teachers continue to "do more" to teach coding, they and their students are likely to get better and improve their understanding of coding.

5. References

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings from Proceedings of the 2012 annual meeting of the American Educational Research Association*, Vancouver, Canada, Citeseer.

Rich, P. (2012). Inside the black box: revealing the process in applying a grounded theory analysis. *Qualitative Report*, 17, 49. Retrieved from https://eric.ed.gov/?id=EJ981472

Rich, P. J., Larsen, R. A., & Mason, S. L. (2020). Measuring teacher beliefs about coding and computational thinking. *Journal of Research on Technology in Education*, 1-21. doi:10.1080/15391523.2020.1771232