Will using blended learning, peer instruction, pair programming and live coding better facilitate teaching programming and data manipulation?

Lene Jung Kjær

Department of Veterinary and Animal Sciences
University of Copenhagen

Background

Programming can be challenging to teach, as learning how to program is often perceived as difficult by students (Sharma et al., 2020). Programming is a relatively new discipline and the learning curve is rather steep causing some people to give up before they gain a working understanding on how to code. Teachers of computer programming are faced with a difficult challenge, as there are no perfect and finite guidelines for how to best teach computer programming (Brown & Wilson, 2018; Sharma et al., 2020). However, as learning to code is a very hands-on process, a hands-on approach may be more effective than a more passive approach (Wilson, 2019).

Blended learning can be defined as “learning which combines online and face-to-face approaches” (Heinze & Procter, 2004), and this method has become more widespread in higher education over the last decade (Alammary et al., 2014). Both Wang et al. (Wang et al., 2005) and Sharma et al. (Sharma et al., 2020) found that a blended learning environment, combining lectures and on-line learning platforms, provides flexibility to both teaching and learning computer programming, as the students can take control of their own learning pace. Sharma et al. (Sharma et al., 2020) included live coding and peer instruction in their in-class lectures, which are both forms of active learning.

Active learning is generally defined as "any instructional method that engages students in the learning process" (Prince, 2004), and has become popular in university teaching (Prince, 2004; Rossi et al., 2021), where traditional lectures and teacher-centred pedagogy is being replaced, and students play an active part in their own learning process (Rossi et al., 2021). When students are allowed to dynamically use what they are taught, they often learn better than by simply watching and listening (Wilson, 2019). Active learning can be implemented using in-lecture quizzes, group discussions, peer instruction and other methods. Peer instruction is generally defined as "an opportunity for peers to discuss ideas or to share answers to questions in an in-class environment" and can improve the students' problem solving skills and understanding (Knight & Brame, 2018). Brown and Wilson (Brown & Wilson, 2018) suggest using peer instruction and in particular pair programming when teaching computer programming. Pair programming is when two programmers, one being a more experienced programmer, share one computer and take turns being the one who types the code and being the one to offer comments and suggestions (Brown & Wilson, 2018). This way partners can help each other and clarify any misconceptions. The less experienced student learns by getting instructions from the more experienced student while the latter learns by explaining and having to think about the concepts (Brown & Wilson, 2018). Research suggests that pair programming can lead to improvement in learning outcomes as well as increased retention (Hanks et al., 2011). Live-coding, where the teacher writes code in front of their learners and encourage them to code along is another way of activating the students (Brown & Wilson, 2018). This method enables a learning-by-doing environment and slows the teacher down for the students to follow along and furthermore encourages questioning along the way (Brown & Wilson, 2018).

The vast diversity in students and their knowledge background and experience can make it difficult to simply adhere to one way of teaching computer programming. Adaptive teaching is a way to change and adapt the teaching to match the students' abilities, thus achieving a learner-centric environment (Haddad & Kalaani, 2014). Adaptive teaching may prevent discouraging students from learning computer programming and may aid in getting them interested and encourage them to continue their learning even after the course is finished (Settle et al., 2014). Adaptive teaching can however be hard to implement, when trying to accommodate the individual students' learning style as well as working with the entire class and still following the syllabus.

In this study, I applied methods to facilitate teaching computer programming in an online course. I applied blended learning, active learning in the form of live coding, quizzes and pair programming and adaptive teaching in order to facilitate a good learning environment both for the students and for me as a teacher.

Materials and methods

The course

I was one of the teachers in the 2021 PhD course "Epidemiology 1: Planning a study" at the Faculty of Health and Medical Sciences at the University of Copenhagen (https://phdcourses.ku.dk/detailkursus.aspx?id=108081&sitepath=SUND). One of the course objectives is to use the program R to manage epidemiological data. The course usually fits 15-18 students, and lasts 2 months, with mostly self-studies as well as 4 campus-based days with lectures and exercises. The "campus-based" sessions are not mandatory and due to COVID-19, the campus-based sessions have been conducted online on Zoom for all of 2021. I taught the R-programming sessions, and as R is both a programming language and a statistical software program, teaching R can be compared to teaching computer programming.

Blended learning and live coding

I was provided lecture material from previous versions of the course. I changed the lectures slightly to accommodate my teaching style and created pre-recorded videos of three lectures teaching the basics of data handling and the program we would be using for this – R. The pre-recorded lectures were uploaded to Absalon (https://absalon.instructure.com/) and the lecture slides were made available as pdf's along with R files with the discussed code. I created a mandatory assignment on Absalon, ensuring that the students had watched the lectures before start of the online R and data management sessions. At each online session start, I gave a short summary of the online lectures and asked for feedback on any difficulties the students had regarding understanding the topic. I followed up on the feedback to make sure that the topic at hand was understood. This blended learning approach allowed me to be more hands-on during the actual sessions and engage the students, and avoided long-winded lectures on a very practical subject.

After the summary of the online lectures and feedback, I systematically went through the R code used in the online lectures, thus the first 3 days, I showed the student codes taught in each of the three online lectures respectively. I started out by showing the students how to open R and what the different functionalities were before moving on to the actual coding. Then I asked the students to code along with me, so they could experience first-hand what the code did and how to use it. I asked the students to make predictions of possible outcomes of some of the coding, and encouraged the students to ask questions along the way. The questions and responses from the students allowed me to use adaptive teaching, as I could chose to change the course of action or slow down my teaching depending on student feedback. I furthermore prepared topic-relevant quizzes for them to activate and motivate them during the live coding sessions. The quizzes were also used as a form of adaptive teaching, as I could gauge how much of the material the students understood and if I should change my teaching accordingly. The very last day, I asked the students to evaluate and give feedback regarding the structure of the course using Padlet.

Peer instruction/pair programming

Before course start, I again created a mandatory assignment on Absalon where the students had to write a summary of their data handling experience and experience with programs and computer programming. The goal was to use this information to divide the students into pairs, where at least one of them had some experience with programming and statistical programs; a goal that was communicated to the students. The goal was furthermore to gauge, where the students were at knowledge-wise so I could adjust my teaching according to their level (adaptive teaching). On each day, after the live coding, the students would go into Zoom break-out-rooms and do exercises in their pair-programming groups where their newly found R skills could be put to practise on epidemiological data. I made sure to give instructions on the pair programming process to avoid that the more experienced students took over the entire coding exercise. I would continuously oversee these break-out-rooms and join a group, if the students had questions or needed help. These exercises were from previous versions of the course, and were specifically created to fit the teachings for that particular session. The exercises are highly relevant to the topic and are case-based to simulate real-life examples of what the students may encounter when handling epidemiological data. Each following day, I would review the exercises with the students, answer questions, and address potential problems or difficulties to ensure that all students have understood the exercises and the solutions. This review allowed me to use adaptive teaching by changing the pace or structure depending on feedback from the students. The solutions to the exercises along with the associated R-scripts were freely available on Absalon for the students to download. The final and 4th day was spent reviewing what they had learned and doing exercises. After each session, I asked the students to evaluate the pair programming sessions using Padlet (https://padlet.com).

Results

Blended learning and live coding

All students viewed the online pre-recorded lectures before course start, and the feedback received indicated that the students found the pre-recorded lectures and the structure of the sessions useful (Appendix A). There were a few problems with some students not having two screens or not being able to split screens in order to follow the live coding and coding themselves at the same time. I realized that this is something we may want to specify for future versions of the course – particularly if the course is held online instead of on-campus. As previously mentioned, I encouraged the students to ask questions during the live coding sessions, which they embraced. I realized that they had many questions, and that I had to slow down the live coding and explanations immensely. This resulted in me not being able to use all the quizzes prepared, but I found it was better to thoroughly make sure, that all the coding was understood by the students.

Peer instruction/pair programming

Having divided the students into pairs, based on the answers to their assignment, I learned that some of the student answers were not entirely realistic. Some students, who wrote that they had worked with the program R before, had very limited skills and some of these students even ended up being the ones who needed the most help. The first day of teaching and pair programming was therefore quite chaotic, with negative feedback (Appendix

B, comments in the first column) and I realized that I needed to change the course of action. This also meant that I did not have enough experienced students to continue the planned pair programming; instead, I had to convert my approach to group programming, where groups consisted of up to four people including at least one experienced coder. Furthermore, as the online lectures and exercises were not mandatory, I had to manage students regularly dropping in and out of the sessions. This meant that I could not do what I had originally envisioned with the pair programming – I could not have the same people in a group throughout the course, and every day, I had to reconfigure the groups, to have at least one experienced coder among them. This demanded a lot of time and organisation skills.

Three of the students, were very experienced coders and users of the program R. I therefore emailed them before course start to ask if they preferred to work on their own or join the pair programming process. I did this to accommodate their needs and expectations for the course, as the level of programming taught in this course was fairly low and I wanted to avoid them feeling that they were wasting their time. Two of these students preferred to work on their own while one of them did not mind joining the pair programming (later becoming group programming) process. Originally, I let all three of them work on their own, but due to lack of experienced coders for the pair/group programming sessions, I drew on this one experienced student to manage a group of inexperienced coders. Despite starting difficulties and having to change pair programming to group programming, the feedback indicated that these sessions were helpful for the students in learning R (Appendix A and B).

Discussion

I attempted to better facilitate teaching computer programming by implementing blended learning, active learning, peer instruction in the form of pair programming and adaptive teaching by adjusting the teaching to the students and their abilities.

Despite the relatively positive feedback from the students, there were considerate problems regarding both the live coding and the pair programming sessions. Live coding on an online platform, requires the students to either have two monitors or to know how to split the screen in order to both follow the live coding and coding along with the teacher. Some of the students in my class were unable to do so, thus rendering the live coding less useful. This may not be a problem in physical teaching as the live coding can be viewed using a projector while the students code on their own laptops. Problems with the pair programming were mostly due to not having enough experienced programmers in the class as well as the class not being mandatory, resulting in students dropping in and out of the sessions. This interfered with the planned pairs (and subsequently groups), and required me to reconfigure groups on every new day of the course. The learning outcome of the course in regards to R programming may have been further impacted by the short duration of the on-campus (online) sessions. Learning computer programming can be hard and require a lot of practise, thus four online instructional sessions may not be enough for learning. However, the short R-sessions may have inspired the students, shown them the utility of the program, and hopefully opened their eyes to pursue further learning. It should be noted that the student programming summaries may have been truthful in the students' own eyes and that they may have overestimated their own abilities. Objectively quantifying programming ability may not be straightforward and we may often measure student confidence rather than ability, which is important to acknowledge and take into account.

The students responded well to the blended learning approach, with pre-recorded lectures to be viewed before course start. This agrees well with the findings of Wang et al. (Wang et al., 2005) and Sharma et al. (Sharma et al., 2020) that blended learning proved to be a more effective approach when teaching computer programming. The blended approach allowed me to implement active learning during the face-to-face online sessions, where instead of lectures, I asked the students to live code with me, all the while answering whatever questions they had. This way the session consisted of dialogues between the students and I, and the students took an active role in their learning experience. This approach, however, was time consuming and I had to adapt by limiting the extent of the live coding, as more time than expected was used explaining the code. Furthermore, I was not able to use all the quizzes I had prepared for the students due to time restraints. This taught me that using active teaching the way I did, is a trade-off between the planned program (and potentially syllabus) and the methods used to teach. I found it more important to ensure that the teaching material was understood, than sticking strictly to the planned program, however, this is something that needs to be recognized when planning a teaching session.

The live coding sessions along with doing exercises, allowed the students to dynamically use what they were taught, creating a learning-by-doing environment that has been known to strengthen the learning of computer programming (Brown & Wilson, 2018; Wilson, 2019). According to the student feedback, the students found the exercises very useful, particularly that the solutions to the exercises were provided for them to go through at their own pace. Using adaptive teaching by including student abilities in the pair/group programming process, posed a didactic challenge and was harder than expected as trying to meet all the students' needs was very time consuming and logistically difficult. I tried using adaptive teaching at the individual level through pair programming, carefully dividing the students into compatible pairs (learning wise). However, when the pair programming did not work, due to the limited number of experienced coders, I tried to apply adaptive teaching through a group programming approach, where at least one experienced coder was assigned to each group. This seemed to work really well, and the student feedback was mostly positive – some of the students even commented that they liked the group structure better than working in pairs, as they had input from more people when solving the exercises. One of the very experienced coders that took part in the group programming exercises conveyed, that they also benefitted from teaching and explaining the code to other students, which was exactly the reasoning behind this approach (Brown & Wilson, 2018). The group programming experience, however, showed me how this method relies heavily on the availability of experienced students, and thus the learning process of the students can be highly dependent on their colleagues. This dependence may be more pronounced regarding hands-on topics such as computer programming, where the students have to actively use their knowledge oftentimes directly following or during teaching sessions. The availability of experienced students may also become more important for short-term courses, where the teacher cannot follow the students progression over a longer time period and thereby recognize more skilled students that could be drawn on to help other students. This dependence on experienced students seems more pronounced for an online setting. Since this first experiment, I have taught the same course in 2022 with the same setup as described here, but where the on-campus days were actually physical. In a physical setting, I could continuously gauge the students and their learning, listen in on their in-group conversations and answer group questions in plenum, which is difficult to do online when the students are isolated into breakout rooms. Thus, in a physical setting, there may be less need for experienced coders due to more student-teacher interaction.

The results from this project has inspired me to continue using blended learning and active learning in the form of live coding and pair/group programming to teach computer programming, which I have done in the physical 2022 course. Overall, the students provided mostly positive feedback for the session structure, so I was happy to see, that this way of teaching was helpful for the students. If this course is taught online again, I need to make it clearer to the students that they need two screens or split screens to follow the live coding. I also need to think of ways to correctly gauge the students' programming abilities to avoid relying on their own, sometimes overconfident, assessments. I would start out using group programming instead of pair programming. I used group programming for the 2022 physical course, and it seemed to work really well for the students as I only received positive feedback regarding the group work. Furthermore, if possible, I would like the on-campus (online) sessions to be mandatory, so that the students can stay in the same groups throughout the week.

Conclusion

The applied approaches - blended learning, live coding and pair/group programming – garnered positive feedback from most of the students, and seemed to encourage the students to learn the program R, thereby making the students take an active part in their own learning process. However, it is important to note that incorporating dialogue and activation of the students may create a trade-off between teaching the syllabus and time spent incorporating the different approaches and activities. Computer programming is a difficult topic to teach, and despite the availability of numerous different approaches, teachers of computer programming are still faced with difficult challenges.

References

Alammary, A., Sheard, J., & Carbone, A. (2014). Blended learning in higher education: Three different design approaches. *Australas. J. Educ. Technol*, *30*, 440–454. https://doi.org/10.14742/ajet.693

Brown, N., & Wilson, G. (2018). Ten quick tips for teaching programming. *PLoS Comput. Biol*, *14*, 1–8. https://doi.org/10.1371/journal.pcbi.1006023

Haddad, R., & Kalaani, Y. (2014). *Adaptive teaching: An effective approach for learner-centered classrooms*. American Society for Engineering Education, International Forum. Indianapolis. https://doi.org/10.18260/1-2--17166

Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: A literature review. https://doi.org/10.1080/08993408.2011.579808

Heinze, A., & Procter, C. (2004). Reflections on the use of blended learning. *Reflections on the use of blended learning* (pp. 1–13). University of Salford.

Knight, J., & Brame, C. (2018). Peer instruction. *CBE Life Sci. Educ*, *17*, 1–4. https://doi.org/10.1187/CBE.18-02-0025/ASSET/IMAGES/LARGE/CBE-17-FE5-G002.JPEG

Prince, M. (2004). Does active learning work? a review of the research. *Journal of Engineering Education*, *93*(3), 223–231.

Rossi, I. V., de Lima, J. D., Sabatke, B., Nunes, M. A. F., Ramirez, G. E., & Ramirez, M. I. (2021). Active learning tools improve the learning outcomes, scientific attitude, and critical thinking in higher education: Experiences in an online course during the covid-19 pandemic. *Biochemistry and Molecular Biology Education*, *49*(6), 888–903.

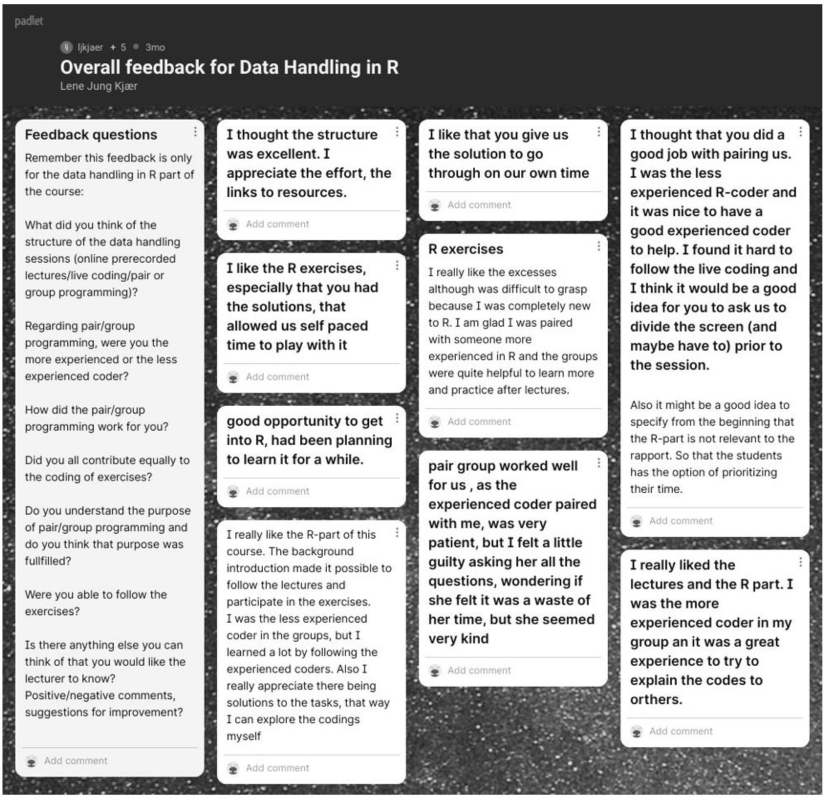
Settle, A., Vihavainen, A., & Miller, C. (2014). Research directions for teaching programming online [PDF].

Sharma, M., Biros, D., Ayyalasomayajula, S., & Dalal, N. (2020). Teaching programming to the postmillennial generation: Pedagogic considerations for an is course. *J. Inf. Syst. Educ*, *31*, 96–105.

Wang, F., Fong, J., Choy, M., & Wong, T.-L. (2005). Blended teaching and learning of computer programming. In H. Leung, F. Li, R. Lau, & Q. Li (Eds.), *Advances in web based learning icwl 2007* (pp. 606–617).

Wilson, G. (2019). Ten quick tips for delivering programming lessons. *PLoS Comput. Biol*, *15*, 1–7. https://doi.org/10.1371/journal.pcbi.1007433

A Overall feedback to programming session structure



padlet

ljkjaer + 5 • 3mo

Overall feedback for Data Handling in R

Lene Jung Kjær

Feedback questions

Remember this feedback is only for the data handling in R part of the course:

What did you think of the structure of the data handling sessions (online prerecorded lectures/live coding/pair or group programming)?

Regarding pair/group programming, were you the more experienced or the less experienced coder?

How did the pair/group programming work for you?

Did you all contribute equally to the coding of exercises?

Do you understand the purpose of pair/group programming and do you think that purpose was fullfilled?

Were you able to follow the exercises?

Is there anything else you can think of that you would like the lecturer to know? Positive/negative comments, suggestions for improvement?

Add comment

I thought the structure was excellent. I appreciate the effort, the links to resources.

Add comment

I like the R exercises, especially that you had the solutions, that allowed us self paced time to play with it

Add comment

good opportunity to get into R, had been planning to learn it for a while.

Add comment

I really like the R-part of this course. The background introduction made it possible to follow the lectures and participate in the exercises. I was the less experienced coder in the groups, but I learned a lot by following the experienced coders. Also I really appreciate there being solutions to the tasks, that way I can explore the codings myself

Add comment

I like that you give us the solution to go through on our own time

Add comment

R exercises

I really like the excesses although was difficult to grasp because I was completely new to R. I am glad I was paired with someone more experienced in R and the groups were quite helpful to learn more and practice after lectures.

Add comment

pair group worked well for us , as the experienced coder paired with me, was very patient, but I felt a little guilty asking her all the questions, wondering if she felt it was a waste of her time, but she seemed very kind

Add comment

I thought that you did a good job with pairing us. I was the less experienced R-coder and it was nice to have a good experienced coder to help. I found it hard to follow the live coding and I think it would be a good idea for you to ask us to divide the screen (and maybe have to) prior to the session.

Also it might be a good idea to specify from the beginning that the R-part is not relevant to the rapport. So that the students has the option of prioritizing their time.

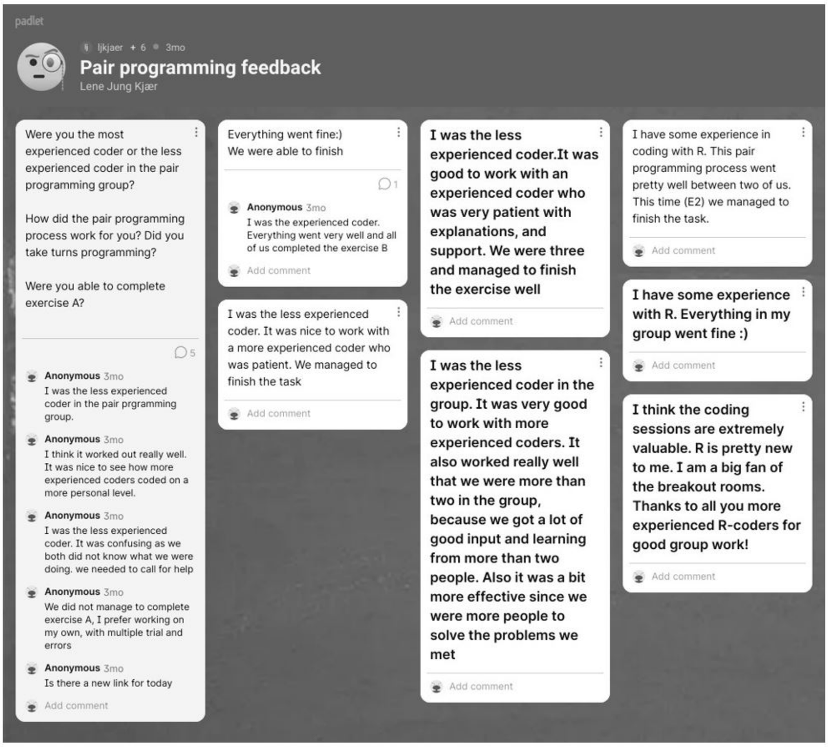
Add comment

I really liked the lectures and the R part. I was the more experienced coder in my group an it was a great experience to try to explain the codes to orthers.

Add comment

Picture exported from Padlet

B Feedback on the pair programming


padlet

ljkjaer + 6 • 3mo

Pair programming feedback

Lene Jung Kjær

Were you the most experienced coder or the less experienced coder in the pair programming group?

How did the pair programming process work for you? Did you take turns programming?

Were you able to complete exercise A?

5

Anonymous 3mo
I was the less experienced coder in the pair prgramming group.

Anonymous 3mo
I think it worked out really well. It was nice to see how more experienced coders coded on a more personal level.

Anonymous 3mo
I was the less experienced coder. It was confusing as we both did not know what we were doing. we needed to call for help

Anonymous 3mo
We did not manage to complete exercise A, I prefer working on my own, with multiple trial and errors

Anonymous 3mo
Is there a new link for today

Add comment

Everything went fine:)
We were able to finish

1

Anonymous 3mo
I was the experienced coder. Everything went very well and all of us completed the exercise B

Add comment

I was the less experienced coder. It was nice to work with a more experienced coder who was patient. We managed to finish the task

Add comment

I was the less experienced coder.It was good to work with an experienced coder who was very patient with explanations, and support. We were three and managed to finish the exercise well

Add comment

I was the less experienced coder in the group. It was very good to work with more experienced coders. It also worked really well that we were more than two in the group, because we got a lot of good input and learning from more than two people. Also it was a bit more effective since we were more people to solve the problems we met

Add comment

I have some experience in coding with R. This pair programming process went pretty well between two of us. This time (E2) we managed to finish the task.

Add comment

I have some experience with R. Everything in my group went fine :)

Add comment

I think the coding sessions are extremely valuable. R is pretty new to me. I am a big fan of the breakout rooms. Thanks to all you more experienced R-coders for good group work!

Add comment

Picture exported from Padlet