

# Should we still teach or learn coding? A postgraduate student perspective on the use of large language models for coding in ecology and evolution

by Campbell, H., Bluck, T., Curry, E., Harris, D., Pike, B. and Wright, B.

Copyright, publisher and additional information: Publishers' version distributed under the terms of the [Creative Commons Attribution License](#)

[DOI link to the version of record on the publisher's site](#)



**Harper Adams  
University**

Campbell, H., Bluck, T., Curry, E., Harris, D., Pike, B. and Wright, B. (2024) 'Should we still teach or learn coding? A postgraduate student perspective on the use of large language models for coding in ecology and evolution', *Methods in Ecology and Evolution*, 15(10), pp.1767-1770.

## FORUM

# Should we still teach or learn coding? A postgraduate student perspective on the use of large language models for coding in ecology and evolution

Heather Campbell  | Thomas Bluck | Ella Curry | Derrick Harris | Billie Pike |  
Bethany Wright

Centre for Crop and Environmental  
Science, Agriculture and Environment  
Department, Harper Adams University,  
Newport, UK

**Correspondence**

Heather Campbell

Email: [hcampbell@harper-adams.ac.uk](mailto:hcampbell@harper-adams.ac.uk)

**Handling Editor:** Robert B. O'Hara

**Abstract**

1. The extent to which coding skills are taught within ecology and evolution curricula remains largely unquantified. While coding, and especially R, proficiency is increasingly demanded in academic and professional contexts, many students encounter coding for the first time as postgraduates, presenting a steep learning curve alongside learning advanced statistics.
2. With the emergence of large language models (LLMs), questions arise regarding the relevance of teaching coding when many of these tasks can now be automated. Here, we explore students' experiences with using LLMs for coding, highlighting both benefits and limitations.
3. Through qualitative analysis of student perspectives, we identify several advantages of using LLMs for coding tasks, including enhanced search capabilities, provision of starting points and clear instructions, and troubleshooting support. However, limitations such as a lack of responsiveness to feedback and the prerequisite of extensive prior knowledge pose challenges to the effectiveness of student use of LLMs for coding at a beginner level. Concerns also arise regarding future access to LLMs, potentially exacerbating inequities in education.
4. Despite the potential of LLMs, we argue for the continued importance of teaching coding skills alongside their integration with LLM support. Tutor-supported learning is essential for building foundational knowledge, facilitating comprehension of LLM outputs and fostering students' confidence in their abilities. Moreover, reliance solely on LLMs risks hindering deep learning and comprehension, thereby undermining the educational process. Our experiences underscore the significance of maintaining a balanced approach, leveraging LLMs as supplementary tools rather than substitutes for coding education in ecology and evolution courses.

**KEYWORDS**

artificial intelligence, ChatGPT, coding, education, large language models, teaching

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2024 The Author(s). *Methods in Ecology and Evolution* published by John Wiley & Sons Ltd on behalf of British Ecological Society.

## 1 | INTRODUCTION

There are limited data on how widely coding is taught in undergraduate and postgraduate ecology and evolution courses. R is the most commonly used programming language in ecology publications and proficiency in R is increasingly a requirement listed in job and postgraduate course advertisements (Auker & Barthelmess, 2020). Despite this, anecdotally it seems that many students first encounter coding and using R during postgraduate study. This can be a steep learning curve because students must simultaneously learn advanced statistics as well as a programming language. These are separate but intertwined skills.

Large language models (LLMs) are becoming increasingly reliable at writing code, raising the possibility that coding is no longer a useful skill for students to learn. As recounted in the Perspectives article, Cooper et al. (2024) posed the question to workshop participants of whether we still need to teach coding if LLMs can do it for us. While we do not know the exact demographic of participants at the workshop it is unlikely to have included many taught (as opposed to research) students. We find that most articles on student use of artificial intelligence (AI) are written by academics (examples being Duffy, 2024 for ecology and Ellis & Slade, 2023 for statistics) meaning that the student voice is not represented in the literature. Cooper et al. (2024) advocate for the continued teaching of coding; we agree and in addition to their Perspectives paper we offer our personal student reflections on using LLMs for coding. Our experiences are primarily with using R and ChatGPT, but the lessons apply equally to other coding languages and LLMs. The ideas presented here arose from a discussion on AI that followed an R coding workshop analysing community ecology data. The workshop took place in February 2024 between postgraduate MSc students at Harper Adams University. Quotes are drawn directly from a transcript recording of the discussion and have been anonymised to maintain student privacy.

## 2 | HOW DO STUDENTS USE LLMs FOR CODING?

### 2.1 | A better way to search

Large language models can be more useful than the internet for answering coding questions. Performing web searches often does not yield useful results and websites do not have the exact information required, whereas LLMs give specific answers to individual queries. LLMs can interrogate data in more detail, finding different things and offering greater insight than traditional search engines. Online help forums can be patronising and hostile towards beginners, so we find that LLMs are preferable as they lack this negative aspect. Additionally, they will provide multiple suggestions to try after an initial query is raised which can be helpful in finding solutions.

I always found if I'm trying to research something I just go around websites and pages skipping through

and in the end, I've gone around in circles. So that's helpful [about LLMs].

### 2.2 | Providing a starting point and clear instructions

Large language models are useful for suggesting ideas and pulling together the outline of a script. While they do not provide complete solutions, they will suggest analytical approaches or relevant packages that can be researched in greater detail elsewhere. Another helpful feature is being able to either simplify or expand answers if needed. Our experience of in-person and online tutorials is that statistics instructors do not always speak to non-statisticians in plain language and this can be a barrier to learning. Because LLMs always use plain language, they offer clear explanations that are more user-friendly to beginners. This is especially helpful when working outside of classroom hours when tutors are not available.

You can ask Chat GPT 'Please can you say it more simply' and then 'Say it even more simply' and it will. This really helps if I need to understand a concept where explanations elsewhere (like in forums) are unclear, because I can request the level of understanding I need for that situation.

### 2.3 | Troubleshooting

Large language models can be useful for checking code for syntax errors and providing analysis of error codes. A basic knowledge of R is needed to understand the LLM outputs but this is one of the more valuable and widely used applications of LLMs when learning to code.

I found it useful for detecting syntax errors in my code. If I had a misspelling, missed a comma out or had a bracket out then I could ask ChatGPT to find it. If I get an error message, then I normally start with pasting the message into ChatGPT which will give me information on that error that I can use to figure out the problem.

## 3 | LIMITATIONS OF LLMs FOR CODING

### 3.1 | Lack of responsiveness

While we do believe that LLMs are useful and have their place, there are many difficulties that limit their usability. A repeated theme we have experienced is a lack of responsiveness to feedback. You can request an alternative answer, but it will not always respond to what has been said, instead repeatedly giving the original or a similar answer. Using a second LLM to break the cycle has sometimes been successful, but is scarcely satisfactory.

I found it [ChatGPT] useful, but your request has to be very specific because it does not take feedback well. If it provides you with code that doesn't work and you request alternative code, even if you explain why the previous response did not work, the new response will have the same information.

### 3.2 | Requirement for extensive prior knowledge

Our biggest limitation, and one that stops many of us using LLMs at all, is the requirement for a good working knowledge of coding and statistics in order to provide sufficiently detailed prompts, and to use and interpret AI-generated code. Because of concerns about outputs being incorrect, an inexperienced user could be misled if they relied solely on an LLM to provide information. For example, the 'hallucinations' of LLMs include making up functions and packages in R that do not exist as well as misinterpreting results. This means that a solid foundational knowledge is required to sense check the LLMs outputs.

The big problem I've always found with ChatGPT is if you try to use it as a starting point for interpreting results, then it can say things that are completely wrong. And so, you definitely wouldn't want to say "interpret this" and then just copy and paste in your results. In a recent example, I double checked my interpretation of my results with ChatGPT to see if it agreed with what I'd done, and it just said things which were completely wrong. The interesting thing was, when I said "are you sure that this interpretation is right", it said "Oh, actually, no, I got that wrong. Sorry." And it never got to the right answer. It can get quite confused about the results of any statistical tests even when provided with the test results and a tidy dataset.

I've had it get things wrong, for example when I couldn't work out which package I'd used I gave ChatGPT my code and requested it identify the package used for a specific section out of the listed packages I had installed. The response suggested three different packages, none of which I had used. When told these packages were not applicable, ChatGPT acknowledged it was wrong but would not provide an alternative answer.

### 3.3 | Access to LLMs in the future

We are also concerned that particular LLMs will be tied to individual institutions (Field, 2024) and will move towards paid versions that will not be accessible to all students. This becomes an equity issue if

universities do not provide access since students who can afford a subscription will be at an advantage. This is a particular issue if coding is no longer being taught, and there is a reliance on LLMs that are then needed to complete assignments and research projects.

## 4 | LLMs AND THE TEACHING AND LEARNING PROCESS

We feel strongly that coding still needs to be taught as part of ecology and evolution courses. LLMs can support learning but should not be used as a replacement for teaching. If the teaching process itself was reliant on LLMs, awareness of students' understanding would be solely based on assignment or examination outcomes, with missed opportunities during study to provide help and support for those struggling. Tutor-supported (as opposed to AI-supported) learning is essential for gaining a core understanding of coding, and this in turn helps with understanding coding-related LLM outputs. LLMs can provide guidance once initial knowledge and skills exist but do not provide the necessary foundational knowledge of coding.

It [ChatGPT] definitely doesn't replace a human being, as in someone presenting and teaching. It enhances that.

ChatGPT is useful as a guide to double check concepts outside of the classroom, but not to learn concepts or topics in their entirety.

We feel it is important for students to develop confidence in their own capabilities and to discover that, with help and guidance, they are capable of learning challenging skills that previously seemed intimidating. If students are encouraged to rely solely on LLMs, they will never push certain boundaries of their knowledge and confidence, and risk convincing themselves that there are certain core skills that they 'can't do'. Most importantly, we feel that by relying on LLMs to write code and interpret results, the process of deep learning and comprehension is completely removed. Given that the purpose of education is to gain new skills and understanding, it feels pointless to outsource this process. We live in an increasingly data-rich world. Even as AI capabilities evolve and LLMs become more sophisticated we feel that there will always be a place for students to develop their expertise in any fields related to data handling, interrogation and interpretation.

It's really helpful with coding for checking what you're doing, but other than that, I haven't used it for anything else. I think it cuts out what I see as important parts of the process of doing a lot of things. If I just ask it something and it gives the answer, then I haven't gone through that process of researching, exploring, and interpreting it myself.

Learning to code has been a hugely affirming experience and has given me much greater confidence in my abilities. I came to the MSc from a non-scientific background, and I think it was extremely important for me to find out that I could do this previously impossible-seeming, intimidating skill. If I had just relied on ChatGPT to do all of my coding for me, I would never have found out that I could do it myself. So, I think it's really important to discover that you can learn these skills, and if coding wasn't taught then I'd never have learned that about myself, and I would be a less confident student and scientist.

### AUTHOR CONTRIBUTIONS

The manuscript was conceived by Heather Campbell following a discussion on the use of ChatGPT for coding during an R tutorial in community ecology analysis for an MSc module at Harper Adams University. Students (Thomas Bluck, Ella Curry, Derrick Harris, Billie Pike and Bethany Wright) contributed equally to the discussion and were all invited to co-author the manuscript. Heather Campbell led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication. ChatGPT v.3.5 wrote the abstract structure based on prompts by Heather Campbell who then rewrote the suggested draft into the final version.

### ACKNOWLEDGEMENTS

We would like to thank all the postgraduate students from module C7049 Ecological Entomology at Harper Adams University who participated in the discussion that formed the basis of this commentary but opted out of co-authorship. Many thanks to the reviewer and editor who provided very kind and helpful comments, and who were supportive of publishing work by a group of MSc students writing their first peer-reviewed paper.

### FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

### CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest.

### PEER REVIEW

The peer review history for this article is available at <https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.14396>.

### DATA AVAILABILITY STATEMENT

This manuscript does not contain any data/code.

### ORCID

Heather Campbell  <https://orcid.org/0000-0001-7483-3281>

### REFERENCES

- Auker, L. A., & Barthelmess, E. L. (2020). Teaching R in the undergraduate ecology classroom: Approaches, lessons learned, and recommendations. *Ecosphere*, 11(4), e03060. <https://doi.org/10.1002/ecs2.3060>
- Cooper, N., Clark, A. T., Lecomte, N., Qiao, H., & Ellison, A. M. (2024). Harnessing large language models for coding, teaching, and inclusion to empower research in ecology and evolution. *EcoEvoRxiv*. <https://doi.org/10.32942/X2PS48>
- Duffy, M. (2024). Generative AI and graduate training in ecology. *Dynamic Ecology* blog. <https://dynamicecology.wordpress.com/2024/01/15/generative-ai-graduate-training-in-ecology/#more-66157>
- Ellis, A. R., & Slade, E. (2023). A new era of learning: Considerations for ChatGPT as a tool to enhance statistics and data science education. *Journal of Statistics and Data Science Education*, 31, 128–133. <https://doi.org/10.1080/26939169.2023.2223609>
- Field, H. (2024). OpenAI announces first partnership with a university. CNBC Website. <https://www.cnbc.com/2024/01/18/openai-announces-first-partnership-with-a-university.html>

**How to cite this article:** Campbell, H., Bluck, T., Curry, E., Harris, D., Pike, B., & Wright, B. (2024). Should we still teach or learn coding? A postgraduate student perspective on the use of large language models for coding in ecology and evolution. *Methods in Ecology and Evolution*, 15, 1767–1770. <https://doi.org/10.1111/2041-210X.14396>