ISSN 2057-2069

Education in Practice is an Open Access journal
University of Birmingham, United Kingdom

GenAI: Opportunities and challenges for teaching and learning of coding

Ruth Geen (School of Geography, Earth and Environmental Sciences)

Contact: r.geen@bham.ac.uk

Abstract

Generative AI (GenAI) presents a dramatic change in the landscape of Higher Education and work. Practitioners using computer programming have historically drawn not only on formally published material but also webpages, including forums, for developing syntax. GenAI now offers a further tool for writing code, provided its limitations are understood. This evaluation discusses case studies of current practice in coding education within the School of Geography Earth and Environmental Sciences, alongside a broader review of the literature on the use of GenAI in coding education. Outcomes indicate that students are already making use of GenAI in coding, that it has potential to overcome barriers to learning coding and enhance self-efficacy, but that student GenAI literacy is currently limited. Suggestions for enhancement of practice are provided.

Introduction

ChatGPT launched on 30th November 2022, providing a public interface for OpenAI's generative pre-trained transformer (GPT) models (OpenAI, 2022). ChatGPT's ability to quickly produce apparently plausible answers to questions in correct prose has challenged higher education institutions to rapidly reconsider teaching and assessment strategies (Rudolph *et al.*, 2023). The role GenAI will play in the workplace remains to be seen, and this presents challenges in designing authentic assessment. For a given topic, is it best to incorporate the use of GenAI into the curriculum to train students for a world in which it may be in common use, or to ban its use and return to in-person, invigilated assessments so that students must rely on their own recall and skills?

One area in which GenAI is perceived to be useful is in writing computer code. Here, I evaluate GenAI's (particularly ChatGPT) current usage and potential as a tool for teaching beginner-level coding for data analysis, from my perspective as a BSc and

MSc dissertation supervisor in the School of Geography, Earth and Environmental Science (GEES) at the University of Birmingham.

First, the need for coding education is discussed, alongside challenges in teaching a cohort with a limited computer science background. This is followed by a case study of current teaching practice for two coding modules within GEES, based on perspectives shared by the module leads: an undergraduate module in statistics and R programming taught to Geography, Planning, and Environmental Science students, and an MSc module in statistics, UNIX and R taught to students on Atmospheric Science programmes. Analysis of available assessment grades for the MSc module suggests GenAI may be having an impact on non-invigilated assessment. I then briefly review literature on student perceptions of GenAI, and implementation of GenAI in undergraduate teaching of coding. The evaluation concludes with suggestions for enhancement of practice at Birmingham through the inclusion of GenAI training for coding in connection to the dissertation module. Although the case studies focus on the context of geographers, the outcomes and recommendations are broadly applicable to teaching coding to those without a Computer Science background.

Context: importance of coding in higher education and beyond

The school computer science curriculum was overhauled in 2014. Under current guidance, children should learn the fundamentals of computer programming in primary school, often using a visual programming language such as Scratch (DEI, 2013a), and by the age of 14 should have additionally used a textual programming language such as Python (DEI, 2013b). However, the majority of text-based coding education takes place under the Computer Science GCSE programme, which is only taken by a small proportion of students (The Royal Society, 2017). In addition, schools have struggled to recruit computer science teachers, with recruitment for computing teachers in England falling 64% below target for 2023/2024 (Maisuria et al., 2023). The majority of students therefore generally begin university with minimal experience in text-based programming languages.

The UK has a recognised shortfall in digital skills (e.g. Royal Academy of Engineering, 2023), while globally the World Economic Forum (WEF) emphasises a global surge in job creation linked to big data analytics (WEF, 2023). Much of academic research now makes use of computer programming for data analysis and modelling of systems of interest. However, with computer science separated from other studies at school, students arriving to study their chosen topic do not always anticipate that statistics and coding are likely to form an important part of their journey through university.

This issue is well-illustrated within Geography and Environmental Science. Academic staff teaching into these programmes come from a range of backgrounds, including ecology, hydrology, environmental chemistry, and meteorology, and will generally be applying computer programming in their research. Geography sits at the interface of STEM and humanities, and students are often motivated to study to degree level by an interest in world around them (e.g., Hammond and Healy, 2022). However, they may not be aware of the importance of data analysis skills in advanced study of this field. Many UK students beginning their degree may not have studied computer science since pre-GCSE, and have limited interest and confidence in this area. Similar to mathematics, coding is a known source of anxiety for students, with the perception that it is difficult leading to a self-perpetuating cycle of failure, demotivation and avoidance (Chapman, 2010; Nolan and Bergin, 2016). It is therefore critical to introduce these skills carefully to avoid student disengagement, and efforts to do this have already been made in both the BSc and MSc programmes within GEES (Muller and Kidd, 2014).

Case Study: current delivery of coding modules in GEES and impact of GenAI

Within GEES, at undergraduate level, coding and statistics are incorporated into shared Semester 2 modules for 1st year students in Physical and Human Geography, Planning, and Environmental Science. Students learn the basics of applying R for statistical analysis through 10 1-hour practical sessions, coupled with 10 1-hour lectures on statistical methods. The module is assessed through a quiz partway through the term, which includes interpreting output from code, and a report on analysis of a dataset using R. These skills are later consolidated during students' research for their final-year dissertation projections.

At postgraduate taught level, students on the MScs in *Applied Meteorology and Climatology* and *Air Pollution Management and Control*, are taught statistics, UNIX, and R through a shared module in Semester 1. R is taught through 5 2-hour sessions comprising ~10 mins taught material followed by student work through example problems. UNIX is taught through 2 2-hour sessions, with a formative quiz set in the second session. The summative assessments for the module are a final worksheet assessing the R material covered in the course, and an open-book, in-class exam. For the latter, students may use their course notes, pre-written scripts, and the internet, but invigilation aims to avoid collusion or use of GenAI. Although recent student satisfaction data is not available, Muller and Kidd (2014) reported generally good levels of satisfaction with this module between 2006-2013.

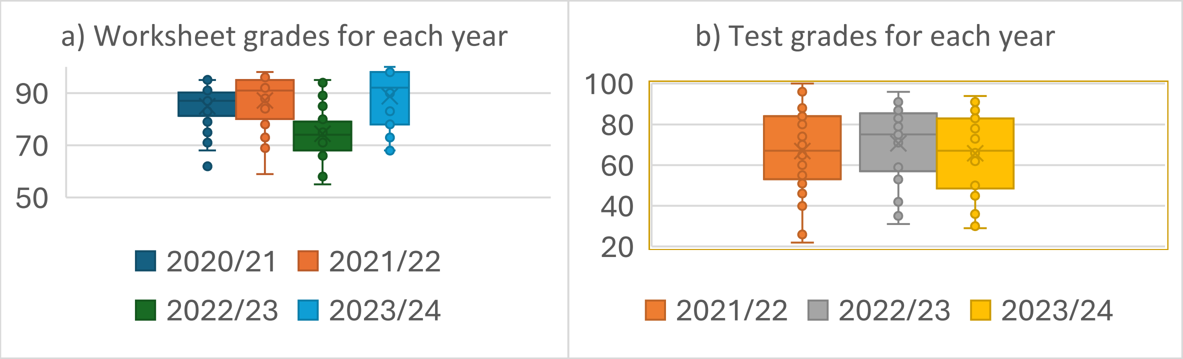
For the undergraduate programmes, GenAI has not been explicitly included in teaching R. However, informal discussions with students in tutorials and dissertation meetings indicate that students are using GenAI to help them develop code for their research projects. At MSc level, GenAI is acknowledged directly in the lectures, and students are suggested to use it for identifying syntax and troubleshooting, although its full benefits and disadvantages are not currently discussed in depth.

Anonymised assessment data from the MSc summative R worksheet was available for 4 academic years from 2020/21 to 2023/2024, and for the in-class test for 3 years from 2021/22 to 2023/24. Deadlines for these assessments are generally in November. As ChatGPT launched on 30th November 2022, it is therefore unlikely that students made significant use of it in their assignments for the 2022/23 sitting, making 2023 the first ‘post-ChatGPT’ year. Average percentage grades for the two assignments are shown in Table 1, and Fig. 1 shows box and whisker plots illustrating the spread of marks for each year. For the worksheet, the average mark over all students for pre-ChatGPT years is 82.3, while the average for the test is 68.7. Unfortunately, no data is available for how GenAI has been used by students in this module, nor how this has affected their perceptions of coding and experience of the course; a study into this is proposed in the conclusion of this article.

Year	Worksheet	In-class test
20/21	85.0	
21/22	87.1	66.8
22/23	74.4	70.7
23/24	88.8	65.9

Average grades for the R assessment components in the MSc module.

Caution must be taken in drawing conclusions from these numbers, with 2020-2022 impacted by teaching changes during the pandemic. The short dataset limits the available pre- and post- ChatGPT baselines, and box and whisker plots for each year suggest the low worksheet grade in 22/23 (74.4) was unusually low, although test grades were more typical. However, it can be noted that the average grade for the at-home summative worksheet in 2023/24 is the highest of the four years for which data is available, while the average grade for the in-class test, which is invigilated to avoid reliance on GenAI, is within the range of previous years. This preliminary data cautiously suggests that GenAI usage may be influencing student marks. Welch's t-test suggests a statistically significant difference between pre- and post-ChatGPT grades for the at-home component (p=0.007), although the small sample size limits confidence in this test, and if 22/23 is excluded differences are reduced (p=0.25). Further years of data and benchmarking of the broader ability of each year's cohort across the programme would be needed for a meaningful analysis. However, changes in assessment format across different academic years limit the number of successive years with similar formats, and hence the data available for comparison.



Box and whisker plots for assessment data in each available year for a) the worksheet and b) the in-class test

Both informal discussions with students, and available data on assessment, indicate that students are already exploring the use of GenAI in developing their code. This is beginning to present a real problem for existing assessment frameworks, with invigilation recently detecting use of GenAI during an in-class test. With no formal education provided in use of GenAI for coding, there is a risk that students with limited GenAI and coding literacy graduate without the skills needed to properly scrutinise GenAI output. The remainder of the article therefore reviews the literature on GenAI literacy and use of GenAI in teaching and learning of coding, identifying relevant insights for developing practice.

GenAI literacy

ChatGPT's free interface is built around GPT-4o mini, which is a large language model (LLM). LLMs are artificial neural networks, trained on large volumes of textual data such that they can generate text by predicting sequences of words when supplied with an input prompt. Learners are often not fully aware of the construction and limitations of GenAI, often anthropomorphising these systems and considering them as capable of human-like thought and intelligence (Bewersdorff *et al*., 2023). I have found this to be reflected in my discussions with both MSc and BSc students. If asked how to explain how GenAI works, responses generally include views of it as a very good search engine, or anthropomorphise it as 'thinking'. Some explicit training on GenAI usage was added to GEES undergraduate tutorial programmes this year, which may help to address these misconceptions. MSc students may benefit from similar training.

As an LLM, ChatGPT can produce seemingly plausible text when provided with a prompt, but its answer is based on statistical pattern recognition developed from its training data, rather than a real thought process. Despite this, research indicates that students have proved liable to take output from ChatGPT as trustworthy, even when it might contradict their intuition. Ding *et al*. (2023) introduced a 'make-up assignment' into an undergraduate physics class. Students were instructed to ask ChatGPT the exam questions they had answered incorrectly, copy ChatGPT's answer into the assignment, and identify whether ChatGPT's results were consistent with their own. If ChatGPT's answer differed from theirs, they were instructed to share their answer to ChatGPT and to report its new response. In their submission, they were asked to explain whether they agreed or disagreed with ChatGPT's answer. Student trust in the AI was then measured based on the percentage of questions for which they agreed with ChatGPT's answers. The study revealed that 19 of the 40 students taking part trusted ChatGPT's answer regardless of its accuracy. When prompted with the student's answer, ChatGPT would sometimes also change its initially correct response. As University GenAI resources continue to develop, this type of exercise may be valuable for encouraging critical thinking around GenAI output.

The short time-span since the launch of ChatGPT means that research is limited, but the results of Ding *et al*. (2023) highlight a risk of misplaced trust, arising from misconceptions about GenAI, leading to reinforcement of scientific misconceptions. This motivates a pressing need to ensure that students are GenAI-literate. The University of Birmingham has now developed resources for students around GenAI literacy, which can be recommended as a good first step. However, given both the opportunities and challenges presented, it is worth further considering how GenAI literacy and skill can be developed in the specific context of coding.

GenAI in teaching and learning of coding

Dissertation students I supervise are already describing GenAI as a 'lifesaver' for debugging code in R, and GenAI support for learning R was even referenced during the student speech at the 2024 GEES and SportEx graduation. There is an emerging body of literature suggesting that GenAI tools could be valuable in teaching coding, if their limitations are properly understood. Becker *et al.* (2023) discuss challenges and opportunities arising from GenAI in the context of introductory programming courses for computer science. The opportunities they note include the potential for students to explore a wide range of example problems and their solutions, and to study alternative approaches to solving the same problem. GenAI can also support review of student code and help in explaining error messages or existing code. Problems include the risk of academic misconduct and the need to clearly define and communicate rules for use of GenAI, and ethical issues linked to attribution and licencing, as well as the risk of over-reliance on tools that are known to sometimes produce incorrect results.

The advantages discussed by Becker *et al*. (2023) link closely to some of the key challenges I have encountered in supporting dissertation students with limited knowledge of coding. To achieve their project aims they need to quickly learn specialised packages which are unfamiliar to them, and it is often difficult for students to interpret error messages from their code. GenAI could be used to provide a starting point from which students can further develop code, avoiding lengthy web searches for syntax (Vaithilingam *et al*., 2022). A test run of ChatGPT indicates that it can correctly answer many of the types of questions emailed to me by MSc dissertation students, providing an option that might reduce frustration for both student and supervisor.

However, just as there is a risk of students over-relying on supervisor input and failing to develop self-efficacy, there is a risk of over-reliance on GenAI. In addition, for some questions I posed to ChatGPT the AI provided an incorrect answer. For example, for the prompt 'please write me code to remove leap-days from a dataset with dimensions latitude, longitude, time', rather than writing code to remove 29/02 from data, ChatGPT produced code that I would expect to remove the entirety of data for each leap year. Blind use of GenAI therefore risks introducing errors versus a careful web search and full understanding of the code used.

Recent research has explored ways to address this challenge, which could be used in designing focussed GenAI-coding activities to support learning in the case study modules discussed above. Yilmaz and Yilmaz (2023) explore the use of ChatGPT as an educational aid. They divided a class of 45 students undertaking a 5-week programming course into an experimental group of 21 students whose training included use of ChatGPT and control group of 24 students following the existing curriculum. Both groups were tested before and after teaching to evaluate their computational thinking, self-efficacy, and learning motivation. The study indicated improvements in all aspects when ChatGPT was used. For computational thinking, use of ChatGPT reduced time on identifying syntax and debugging, with students instead using their time to work out creatively how to explain to ChatGPT what was needed for each step, and piecing together the resulting code to construct their final submission. The reduced time on coding and debugging also allowed for increased self-efficacy, accompanied by increased motivation, with students more able to solve syntax problems with the aid of ChatGPT, providing motivation and confidence in their ability to construct computer programmes.

Yilmaz and Yilmaz (2023) propose guidance for teaching staff on designing programming assignments. The version of ChatGPT used was only able to take in text-based questions. By providing questions in a schematic form, it was possible to ensure students formulate their own requests to ChatGPT, rather than simply copying the question into the input field. This mimics how the software might be used in a research context; whether using Google searches or ChatGPT to identify syntax, the challenge for the researcher is often formulating the right question to find the information needed. In line with this, Denny *et al*. (2024) introduce 'Prompt problems' as a programming exercise in a Computer Science module. Again, problems are posed via images, avoiding students simply copying the problem into an LLM. Students submit an LLM prompt and successful solution of the problem posed, which should be concise, show understanding of the problem, and avoid parroting the question (e.g. by explicitly copying example data). Student evaluations were generally positive, although some were reluctant to use GenAI, or concerned about the future of the programming profession.

Informed by the above, alongside trends in assessment grades seen in the case study modules, and discussion with students during dissertation supervision, I suggest implementing specific GenAI training into either coding or dissertation modules. This would include training on prompt development, followed by training on critical evaluation of the resulting code via annotation and testing of the outputs of each line of code.

Conclusions

Anecdotal evidence based on discussions with students during dissertation meetings and tutorials indicates that GenAI is already being used to support learning. Assessment data from the R components of MSc modules covering coding indicate that grades for the at-home assessed component this year were higher than those of previous years, while the in-class assessment grades remained similar. However, the cause of this cannot be attributed to GenAI without further years of grades and information on students' use of GenAI.

The literature raises both opportunities and challenges from GenAI. AI literacy is vital to ensure students approach output with a critical mind (Ding *et al.*, 2023), but GenAI could be a valuable tool to improve self-efficacy and confidence in learning to code (Becker *et al.*, 2023; Yilmaz and Yilmaz, 2023). This support for achieving self-efficacy may be particularly valuable for learners with high coding anxiety, which, from my experiences of dissertation supervision, appears common in GEES students. In most contexts where coding will be used, practitioners have full access to the internet, and in pre-AI practice would use web forums such as Stack Overflow to identify syntax and troubleshoot. Drawing on such forums already requires critical thinking to understand the possible solutions and avoid poor advice from other users. It therefore seems logical to encourage use of GenAI as a tool within authentic assessment, while clearly demonstrating its limitations.

Motivated by this evaluation of practice, I recommend undertaking work to assess ongoing changes in student outcomes in coding assessments across GEES, student perceptions of GenAI and its abilities, their current usage of it for coding, and how they perceive its impact on their understanding, confidence, and outcomes. This could be achieved via focus groups and Canvas quizzes. I also suggest implementing direct training for students in how to best make use of GenAI for coding, including understanding its limitations. This could draw on methods used by Ding *et al*. (2023) and Yilmaz and Yilmaz (2023) to encourage students to reflect on their understanding of LLMs, and to explore their usage in developing code.

In the long term, computer programming education and assessment will need to adapt to account for GenAI, particularly as it becomes more widely part of standard practice in developing code. Closed-book, invigilated exams do not authentically assess coding skills, and may reinforce anxiety around coding. To train confident practitioners, the critical use of GenAI, among other sources of information, should be supported. The approaches of Yilmaz and Yilmaz (2023) and Denny et *al*. (2024) may provide useful starting points for redesigning computing assessment.

Acknowledgements

Discussion of current practice was informed by useful discussions with Francis Pope and Liz Hamilton. This evaluation formed part of an assignment under the HEFi ‘Developing and Enhancing Your Professional Practice’ course; I thank Jamie Morris and Marios Hadjianastasis for their feedback and support throughout the programme.

References

Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J. and Santos, E. A. (2023) 'Programming Is Hard - Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation', *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pp. 500–506. DOI: https://doi.org/10.1145/3545945.3569759.

Bewersdorff, A., Zhai, X., Roberts, J. and Nerdel, C. (2023) 'Myths, mis- and preconceptions of artificial intelligence: A review of the literature', *Computers and Education: Artificial Intelligence*, 4, 100143. DOI: https://doi.org/10.1016/j.caeai.2023.100143.

Chapman, L. (2010) ‘Dealing with Maths Anxiety: How Do You Teach Mathematics in a Geography Department?', *Journal of Geography in Higher Education*, 34 (2), pp. 205–213. DOI: https://doi.org/10.1080/03098260903208277.

Department for Education (2013a) *Computing programmes of study: Key stages 1 and 2*. Available at: https://assets.publishing.service.gov.uk/media/5a7c576be5274a1b00423213/PRIMARY_national_curriculum_-_Computing.pdf. (Accessed 21 February 2025)

Department for Education (2013b) *Computing programmes of study: Key stages 3 and 4*. Available at: https://assets.publishing.service.gov.uk/media/5a7cb981ed915d682236228d/SECONDARY_national_curriculum_-_Computing.pdf. (Accessed 21 February 2025)

Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A. and Reeves, B. N. (2024) 'Prompt Problems: A New Programming Exercise for the Generative AI Era', *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, pp. 296–302. DOI: https://doi.org/10.1145/3626252.3630909.

Ding, L., Li, T., Jiang, S. and Gapud, A. (2023) 'Students' perceptions of using ChatGPT in a physics class as a virtual tutor', *International Journal of Educational Technology in Higher Education*, 20 (1), 63. DOI: https://doi.org/10.1186/s41239-023-00434-1.

Hammond, L. and Healy, G. (2022) 'Engaging with undergraduate geography students' perspectives on the value of geography to a person's education', *Geography*, 107 (3), pp. 137–144. DOI: https://doi.org/10.1080/00167487.2022.2114164.

Maisuria, A., Roberts, N., Long, R. and Danechi, S. (2023) *Teacher recruitment and retention in England*, House of Commons Library. https://researchbriefings.files.parliament.uk/documents/CBP-7222/CBP-7222.pdf.

Mertala, P., Fagerlund, J. and Calderon, O. (2022) 'Finnish 5th and 6th grade students' pre-instructional conceptions of artificial intelligence (AI) and their implications for AI literacy education', *Computers and Education: Artificial Intelligence*, 3, 100095. DOI: https://doi.org/10.1016/j.caeai.2022.100095.

Muller, C. L. and Kidd, C. (2014) 'Debugging geographers: Teaching programming to non-computer scientists', *Journal of Geography in Higher Education*, 38 (2), pp. 175–192. DOI: https://doi.org/10.1080/03098265.2014.908275.

Nolan, K. and Bergin, S. (2016) 'The role of anxiety when learning to program: A systematic review of the literature', *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, pp. 61–70. DOI: https://doi.org/10.1145/2999541.2999557.

OpenAI. (2022) *Introducing ChatGPT*. Available at: https://openai.com/blog/chatgpt (Accessed 21 February 2025).

Royal Academy of Engineering (2023) *Skills shortage in computing could stop the UK meeting its AI goals*. Available at: https://raeng.org.uk/news/skills-shortage-in-computing-could-prevent-the-uk-from-meeting-its-ai-ambitions-says-royal-academy-of-engineering (Accessed 21 February 2025).

Rudolph, J., Tan, S and Tan, S. (2023) 'ChatGPT: Bullshit spewer or the end of traditional assessments in higher education?', *Journal of Applied Learning and Teaching*, 6 (1), Article 1. DOI: https://doi.org/10.37074/jalt.2023.6.1.9

The Royal Society (2017) *After the reboot: Computing education in UK schools*. Available at: https://royalsociety.org/-/media/policy/projects/computing-education/computing-education-report.pdf (Accessed 21 February 2025).

Vaithilingam, P., Zhang, T. and Glassman, E. L. (2022) 'Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models', *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–7. DOI: https://doi.org/10.1145/3491101.3519665.

WEF (2023) *The Future of Jobs Report 2023*. World Economic Forum. Available at: https://www.weforum.org/reports/the-future-ofjobs-report-2023/. (Accessed 14 February 2025).

Yilmaz, R. and Yilmaz, F. G. K. (2023) ‘The effect of generative artificial intelligence (AI)-based tool use on students’ computational thinking skills, programming self-efficacy and motivation', *Computers and Education: Artificial Intelligence*, 4, 100147. DOI: https://doi.org/10.1016/j.caeai.2023.100147.