Incorporating Generative AI as a Data Analysis Partner in the Meteorology Classroom

Dana E. Veron, Ryan Eagan, Paige Burchell
University of Delaware
Gerard J. Mangone Climate Change Science and Policy Hub

Abstract

Graduates from undergraduate and graduate meteorology, climate, and environmental science programs entering private-sector and government jobs in the environmental monitoring, compliance, and analysis job market must have data analysis and visualization skills. A recent analysis of employers in the Environmental Science and Technology Sector indicated that one of the most sought-after skills is in data analysis and visualization, predominantly using Excel and PowerPoint or programming languages such as Python and R. The field of data analysis and visualization is experiencing a rapid shift towards using generative AI to increase efficiency in data analysis and improve data communication and storytelling. In response to this change, we adapted several data analysis exercises embedded in an upper-level undergraduate, flipped-classroom style course on Climate and Climate Change from utilizing spreadsheet type programs like Excel to working with a generative AI platform like ChatGPT as a partner in developing data-analysis code in Python. This shift not only meets the changing demand in private sector jobs but also enhances our meteorology and climate students' Python programming skills by providing real-world examples while asking them to engage deeply in the process of data analysis. Much like the benefits provided to students learning a programming skill utilizing paired programming, it is anticipated that incorporating generative AI as a partner in code development can lower barriers for those learning coding for the first time and provide more opportunities to rapidly develop confidence in their data analysis skills. In addition, this classroom innovation is designed to encourage students to think critically about their data analysis and visualization goals and tools, assessing whether they understand the resulting analysis and if the analysis provides them insight to the research question posed. The ability to question and critique the process and products of data analysis as generative AI becomes more embedded in meteorological and environmental work is a critical skill for our meteorology, climate and environmental students.

Introduction

A recent analysis of employers in the Environmental Science and Technology Sector indicated that one of the most sought-after skills is in data analysis and visualization, predominantly using Excel (Kim et al. 2021). In response to this feedback, GEOG341-Climate and Climate Change, adopted a data analysis learning goal, and scaffolded student learning through a series of five Excel data analysis exercises using publicly available or faculty-provided, research-based, environmental data sets. The exercises build in complexity and review previous skills learned to strengthen student confidence and competence. These exercises culminate in a Climate Change Data Analysis group project that allows students to test and develop their skills on a climate-change related topic such as Arctic sea ice loss, increasing fires in the Amazon and the Southwest, or migration of benthic species due to climate change.

The Environmental Science, Meteorology, and Climate Science faculty have recently agreed to embed Python programming into their course offerings, as Python proficiency is now becoming a more sought-after skill. However, depending on the student's comfort level with and previous exposure to programming, this can be a large lift for students (Hanks et al. 2011). For many students with limited or no programming experience, the time commitment to the exercise is often dominated by the learning curve of programming and frustration of debugging code, detracting from the learning goal of analysis and interpretation of analysis results.

Previous studies have demonstrated that pair programming, where two individuals will develop and write code together using one workstation, leads to higher confidence in solutions, better code quality, and improvement in learning outcomes for students (Hanks et al. 2011). Part of the pair programming paradigm is the practice of walking through or explaining code line by line to your programming partner to help bring one up to speed on the current code and to help identify code defects (bugs) (Elvira et al. 2023). In our approach, we explore how ChatGPT can play the role of a co-developer or pair programming partner for each student.

Since the release of ChatGPT in 2022, the Generative Artificial Intelligence (GAI) field has grown exponentially with rapid adoption and integration across business domains and technology sectors. Student adoption of GAI has put the spotlight on how GAI is disrupting traditional pedagogical methods presenting a mountain of challenges and opportunities of GAI for higher education. Primary concerns around plagiarism, cheating, and an increase in cognitive offloading that may diminish critical thinking skills play against. To foster a successful integration of GAI in higher education, faculty will have to adapt their pedagogical approach to include AI/data literacy as a key critical thinking component (Kasneci et al. 2023). This project aims to demonstrate this principle and approach with a learning objective of helping students to analyze climate data.

Methodology

Using five data analysis exercises in GEOG341-Climate and Climate Change, we worked with an undergraduate research assistant to explore the process of co-developing Python code for each exercise with ChatGPT. Our undergraduate RA, paired with a PhD student to provide continual guidance and technical support, followed the project plan provided in Figure 1.

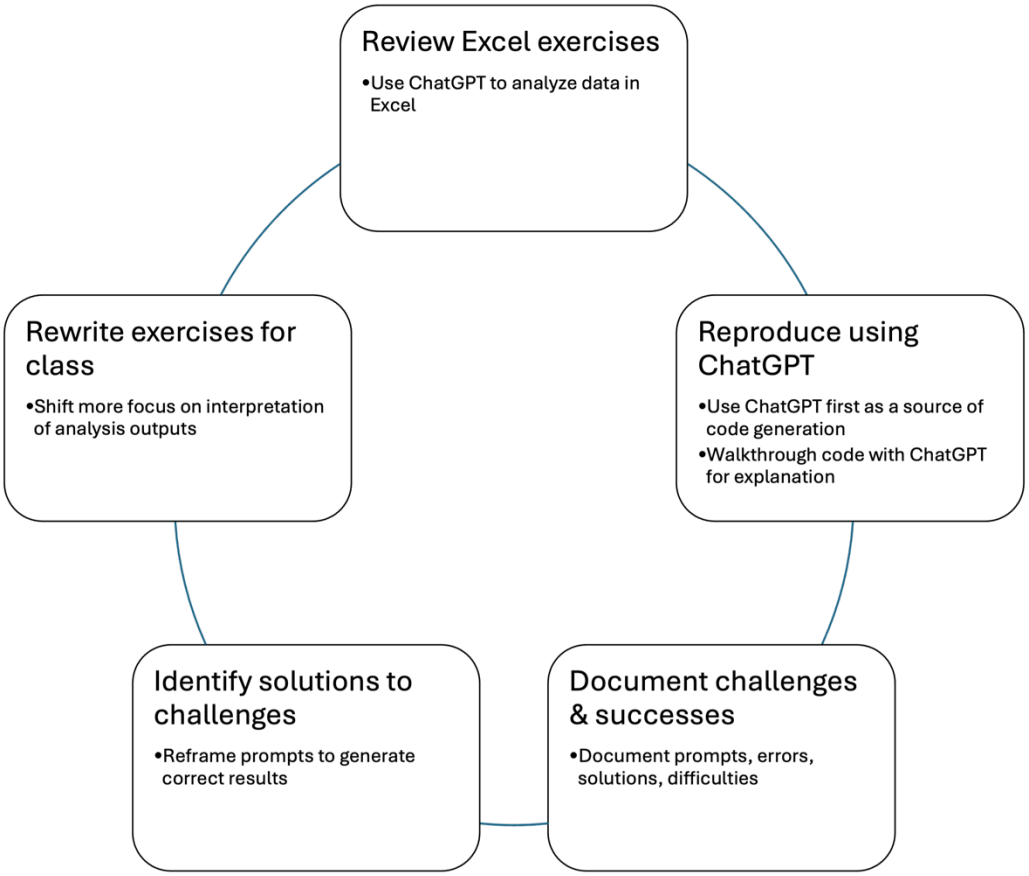


Figure 1: Project Steps. Each assignment was evaluated by our undergraduate RA using Excel, then ChatGPT, documenting the steps throughout. The outputs, steps, and findings were then reviewed with the team to identify and resolve challenges.

For example, the first assignment informs students they are consultants for a wind power company and must perform a wind analysis on the given time series data, which has three columns: sample number, hour and wind speed (m/s). The assignment first asks students to describe the dataset, then goes further to ask specific questions such as determining peak wind gusts for a given period. The other component of the assignment instructs students to generate various plots showing wind speed over time.

Results

We find that prompting becomes a key component of the critical thinking process similar to other findings (Lee et al. 2025), where learning the value of constructing detailed, descriptive prompts and how to clarify and specify within the prompt as needed is critical to achieving the desired outcome. In one instance an initial ask to plot the given dataset resulted in a runtime error trying to decode the provided data.

GPT Prompt by Team	Result	Discussion
How would I plot this dataset in Jupyter Notebook using Python?	Gives a UnicodeDecodeError "**UnicodeDecodeError**: 'utf-8' codec can't decode byte 0xb2 in position 65: invalid start byte	The initial Python code provided
I am receiving the following error: UnicodeDecodeError: 'utf-8' codec can't decode byte 0xb2 in position 65: invalid start byte for the line that says data = pd.read_csv...	This still didn't work, so I added the 'encoding='latin1' ' parameter. However, the graph is showing up blank.	
After using latin1 encoding, I have received a blank graph. The axes are both scaled -0.5 to 0.5. Look at how my file is structured and determine what I should do.	The issue of a blank graph with incorrectly scaled axes is likely due to improper parsing of the file, such as incorrect column alignment or a failure to correctly interpret the separator. Looking at your file structure, here's how you can fix the issue: --- **Analysis of File Structure:** 1. **Separator**: The file seems to use **tabs (`\t`)** as a delimiter. Ensure `sep="\t"` is set when reading the file.	

Figure 2: Excerpt from the prompt and result documentation generated by our RA showing the resultant error after attempting to run the GPT provided code to plot the given dataset. The following two prompts from the RA show the resolution process.

The example in Figure 2 highlights how students can experience GPT's unreliability through broken code outputs, helping them develop troubleshooting skills and a critical approach to AI-generated responses. By rephrasing requests or correcting GPT, students refine their understanding of its limitations while working toward accurate solutions. Coding errors are a common challenge for novice programmers and GPT assists in clarifying cryptic runtime errors, which often consume valuable learning time. In this case, our RA quickly resolved a decoding error with GPT's guidance, demonstrating how AI can be both a source of error and a valuable debugging tool in a low-risk learning environment.

Using the XP/Pair Programming walkthrough paradigm, GPT offers a clear and concise explanation of the provided code, including examples of expected output where applicable, followed by a summary of the overall code. By shifting the focus away from the time-consuming coding and debugging process, GPT allows students to engage more deeply with analysis, interpretation, and the synthesis of ideas and conclusions, enhancing their overall learning experience.

Discussion

Our RA demonstrated that pair programming with ChatGPT dramatically reduced the development time based on her previous programming experiences in other classes/projects using Python. Using GPT to debug runtime errors proved very valuable as this was often an area that tended to occupy a large portion of the assignment timeline.

The ease of producing code that is facilitated by pair programming with ChatGPT could remove the student from the critical thinking necessary to evaluate whether the code is providing the desired analysis. To make sure students engage critically with the material from the start, and to increase data and graphing literacy, homework prompts were added to each assignment that ask the students to explain the purpose of the analysis in their own words and provide handwritten drawings of what they think the result might look like. Once the AI and student co-create a data analysis result, the student is then asked to explain the result based on their understanding of the environment and decide as to whether the result is reasonable. They will also be asked what additional information they would like to know from the data and then produce a result that might give them insight on their question. Students must turn in a log of their prompts and the AI responses.

This project will be delivered through 6 in-class interventions, with asynchronous scaffolding, and synchronous student hours. At the midpoint and the end of the semester, students will be asked for feedback on this project.

References

Elvira, T., T. T. Procko, J. O. Couder, and O. Ochoa, 2023: Digital Rubber Duck: Leveraging Large Language Models for Extreme Programming. *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*, 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), 295–304.

Hanks, B., S. Fitzgerald, R. McCauley, L. Murphy, and C. Zander, 2011: Pair programming in education: a literature review. *Computer Science Education*, **21**, 135–173, https://doi.org/10.1080/08993408.2011.579808.

Kasneci, E., and Coauthors, 2023: ChatGPT for Good? On Opportunities and Challenges of Large Language Models for Education. https://doi.org/10.35542/osf.io/5er8f.

Kim, Y., G. Irvine, and S. Kendus, 2021: *Market Analysis - Master Degree in Environmental Science and Policy_Final.pdf*. University of Delaware, Professional and Continuing Studies,.

Lee, H.-P., A. Sarkar, L. Tankelevitch, I. Drosos, S. Rintel, R. Banks, and N. Wilson, 2025: The Impact of Generative AI on Critical Thinking: Self-Reported Reductions in Cognitive Effort and Confidence Effects From a Survey of Knowledge Workers. CHI Conference on Human Factors in Computing Systems (CHI '25), Yokohama, Japan, ACM, 23.

Acknowledgements
Funding provided by the Paul J Rickards, Jr. Teaching Innovation Grant.