Student Perceptions of GAI Use when Learning Coding

Kellie Keeling and Andrew Urbaczewski

Business Information & Analytics Department

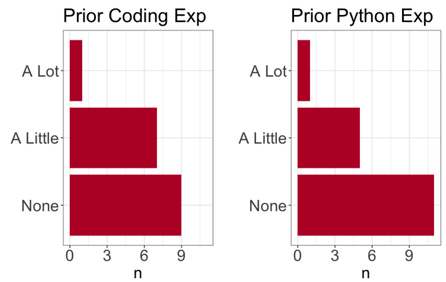
INTRODUCTION

While teaching an introduction to coding course – typically taken as the first or second course in the Business Analytics major or minor - we sought to explore students' perceptions of their use of Generative AI (GAI) tools. Given the widespread accessibility of GAI and its demonstrated accuracy in generating code, we chose to embrace its use in the classroom. However, we were curious to understand:

How did students feel about their use of GAI?

MATERIALS & METHODS

Participants:
17 students in the Fall 24 course who consented. One student self reported a lot of Python and coding experience, but most students had no previous coding or Python experience.



Materials: The original course materials were used including the following topics:

Course Topics
Intro to Python/Google Colab, Python Basics, User Input and Output
Basic Control Logic Using Python/Pseudocoding
Data Structures and Pandas Reading in Data
Functions, Cleaning, and Creating Summaries
Creating Summaries and Merging/Grouping Data
DataFrame Audit and Creating Graphs
Basic Statistics (t tests, ANOVA, Regression)
Good Spreadsheets Modeling/Excel Basics & Built in Functions
Structured Programming with VBA/Control Logic/Using Buttons/Message and Input Boxes/Worksheet Ranges/Charts
VBA Coding Wrap up and Excel/Python Communication
Project create Python program with menu to analyze data/Excel Workbook to create descriptive summaries and bar chart - Presentations to Me

This included an introductory survey with self-reported major and minor and prior experience in coding or Python as None, A Little, or A Lot. There was also an open-ended question: Do you have any comments you would like me to know? This could include anything such as excitement or concerns about the course, strategies that professors have used in the past that have helped you learn, health concerns for you or a family member, disability accommodations, etc. In addition, a "Pre Class" Quiz was administered for participation points on the first day to assess their prior coding and Python aptitude. A sample question is shown below:

```
x=1
while x <= 4:
    print(x)
    x = x + 1
```

What is the output for this code?

A.	1	2	3	4
B.	1	2	3	4
C.	2	3	4	5
D.	2	3	4	

MATERIALS & METHODS

As part of the course assignments, we added reflections on their use of GAI. Here is a sample assignment:

1.Create Pseudocode WITHOUT Generative AI
Create Pseudocode to solve this Prompt with INPUTS, OUTPUTS, PROCESSING, CHECKS
PROMPT: **Calculate a running sum.** Ask a user to enter a number. Keep asking for numbers until a negative number is encountered. When that happens, print the running sum of previous numbers.

2.Create Solution WITHOUT Generative AI
Using the Pseudocode you created and class notes/Python syntax file, solve the question in the prompt.

3.Create Solution WITH Generative AI
Enter the Prompt into a GAI tool and copy its solution below. If you provide any additional prompts to refine the answer, include those and the revised answer as well.

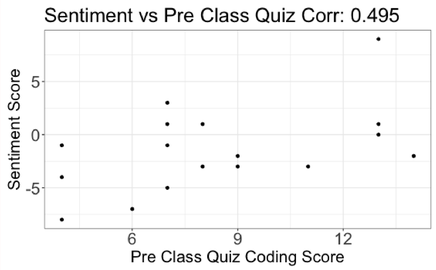
4.Revise your solution from Part 2.
Copy your answer to the problem without GAI from Part 2 below.
Revise your answer based on what you learned from the GAI Answer.
Reflect on how you changed what you did originally.

In all there were a total of 12 assignment questions that asked the students to comment and reflect on their use of GAI with questions similar to the following:

How did GAI perform in [this assignment]?
Do you think your use of GAI helped you with [this assignment]?
In addition, 3 questions were included in the Final Exam:
Reflect on your use of Generative AI in this course
- **How do you think it helped you?**
- **How do you think it hindered you?**
- **How might you use Generative AI as part of your profession or career?**

Procedures & Design: For this study, we collected the students' classification, self-reported prior Python and coding experience, Pre Class Quiz scores, and the text of their reflections. In addition, we captured students mention of the words Excited, Interested, Nervous, or Intimidated in their introductory survey about the course. We used NVIVO to code topics for their answers to the 3 final exam reflections and also calculated an overall sentiment score (0 would be neutral) for all their text comments/reflections. We compared the sentiment scores to the categories from the introductory quiz and also their Pre Class Quiz scores.

RESULTS



FINAL EXAM REFLECTION
How do you think GAI Helped You?

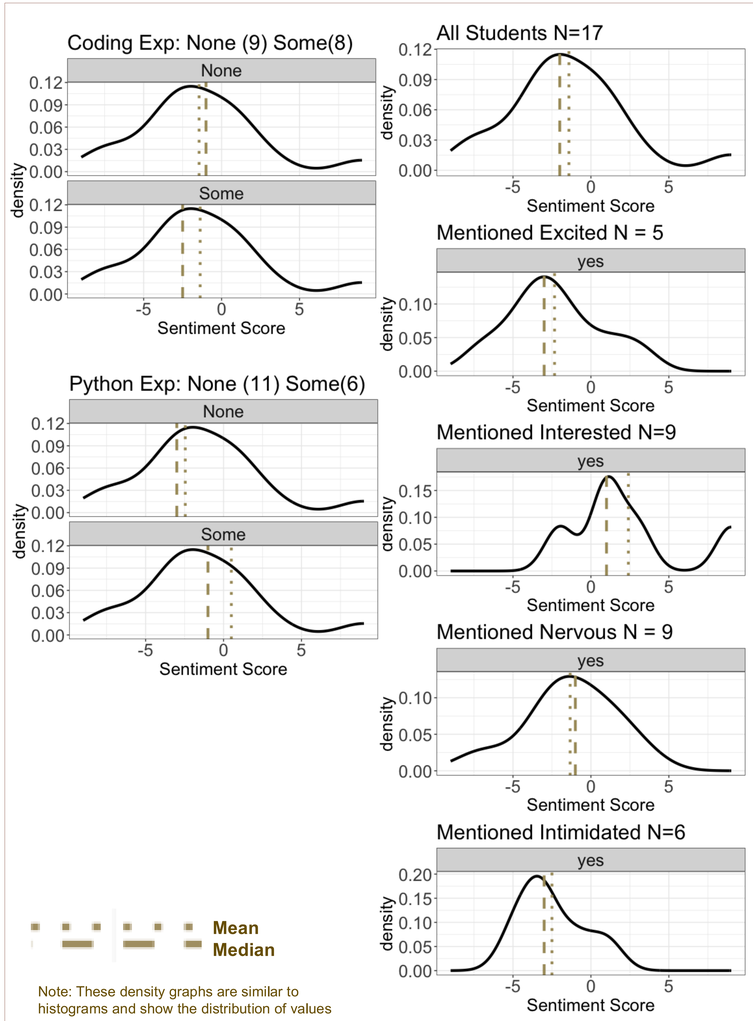
No Programming Exp	Some Programming Exp
Error Explanation	Time Efficiency
Advanced Code Output	Language Flexibility
Overall Helpfulness	Focus on Understanding
Learning Aid	Limited Use in Exams
Effective Use	Efficiency and Guidance
Tool Utilized	

How do you think GAI Hindered You?

No Programming Exp	Some Programming Exp
Understanding Issues	Over-Reliance
Incorrect Outputs	Premature Assistance
Over-Reliance	Learning Impact
Learning Curve	Syntax Memorization
Course-Specific Impact	Future Concerns

How Use GAI in Professional Future

No Programming Exp	Some Programming Exp
Professional Applications	Documentation and Simple Tasks
Perspective and Problem-Solving	Workforce Experience
Efficiency in Basic Tasks	Decision-Making
Limitations and Ethical Considerations	Medical Applications
Programming Assistance	Educational Benefits
Selective Use	



Note: These density graphs are similar to histograms and show the distribution of values

YOU CAN PARTICIPATE

https://bit.ly/KKPresentationFiles

- Digital copy of this poster
- Padlet to answer the Question below:

How do you ensure students are still using their Critical Thinking Skills when they can easily "get an answer" to most questions from GAI?

CONCLUSIONS

Examining the scatterplot as well as the All Students Density plot, we can see that overall sentiment scores were less than 0 (more negative that positive). This doesn't necessarily mean that students don't think positively about GAI but could mean they focused more on problems with using GAI, such as confusion when it provided code that we hadn't covered during class. The correlation between the Pre Class Coding Quiz and overall Sentiment Score was +0.5 indicating some positive correlation, but it was skewed by a single student with a very high sentiment score. Without this student the correlation is .371.

Examining student reflections at the end of the class, students who had no programming experience praised the positives of error explanation and syntax creation, but also noted a tendency to overrely on the tool. Students who had more programming experience focused on the positives of help moving from one programming language to another and also the time efficiency but noted concerns with how it tried to give premature assistance. In discussing future use, overall the students felt that GAI will be a required tool for their success, and they noted that there were many applications where it could be used.

If we examine sentiment scores based on initial coding and programming experience, there is not much difference based on coding experience, but those that had prior Python experience showed a more positive sentiment of GAI. The sentiment based on initial thoughts about the course was less definitive with Excited and Nervous, but the use of the word Interested showed a higher sentiment and use of the word Intimidated showed a lower sentiment.

Takeaways

- Encourage students to not immediately get help from GAI, but instead to write pseudocode first to break down problems and use GAI in chunks rather that just entering an entire assignment prompt.
- The professor had students come to the office to orally present and explain the code in their final project for about 12 minutes. This allowed for probing questions – particularly about code or approaches that we didn't cover in class. The professor observed a notably higher level of understanding among students than anticipated, suggesting that oral demonstrations can be an effective strategy for enhancing code literacy. But about ¼ of the students struggled with explaining some part of their code that either they should have learned or that was probably too advanced.
- With the easy access of GAI tools, professors can spend less time focusing on syntax. The time spent in office hours with debugging issues has almost disappeared, which may also encourage faculty to determine how else to encourage student engagement with faculty during office hours.

Limitations: It is noted that this study included a small sample size with only 17 students. The class was also biased towards Business Analytics Minors.

FUTURE DIRECTIONS

For the Fall 25 version of the course, 3 graded Coding Check In closed notes quizzes have been added. The progressive quizzes are designed to encourage students to think about how code works. We feel this will increase their ability to error check their own work in the future. In addition, there is an increase in demonstrations in class that focus on planning and pseudocoding. This study could also be expanded to explore attitudes about GAI at the beginning of class compared to end of class.

REFERENCES

A portion of this analysis is contained in:
"G(AI)etting with the program: Teaching IS and Analytics in the age of GAI." Urbaczewski, A., Keeling, K. B. (2025). *Communications of the Association for Information Systems*. 56, 1104-1117. https://doi.org/10.17705/1CAIS.05642