AI Tools for Programming Analytics Tasks

Vivek ***Yadav**, Cambridge Systematics*

Suzanne ***Childress**, MTC*

Flavia ***Tsang**, MTC*

September 2025

SESSION FLOW

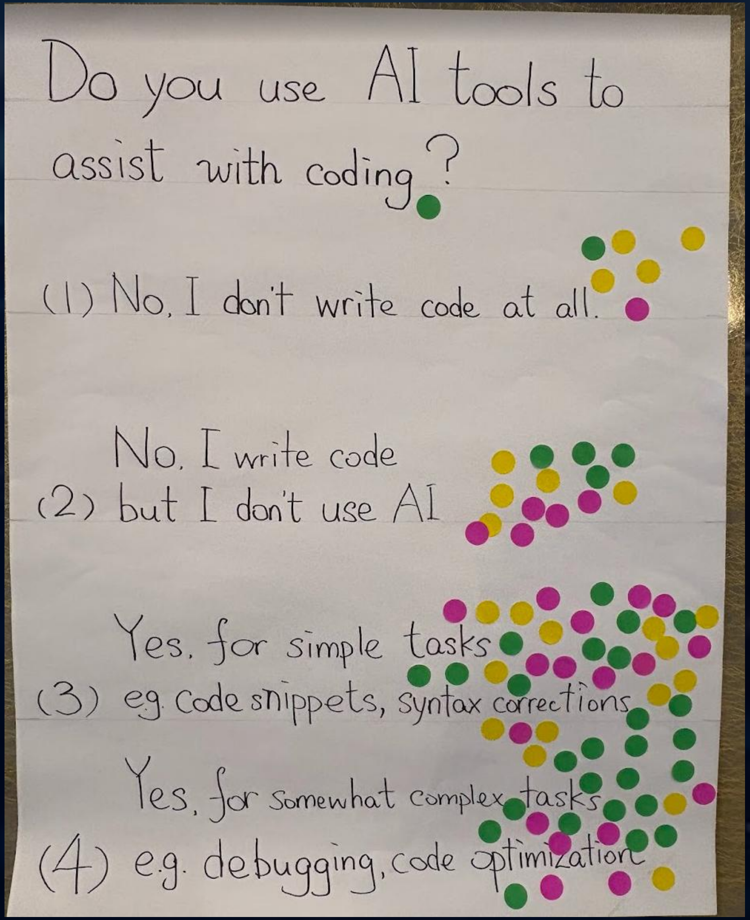
This session will be recorded and summarized by AI.

1. **Welcome** /Introduction (3 min)
2. **Results from TRB Session** at ADB50 about AI(Flavia) (5 min)
3. **Open Discussion**
   - (40 minutes) Suzanne
   - 30 minutes small groups, 10 minutes report and summarize
4. **AI Tools** for Code Completion Demo:
   - (40 minutes) Vivek
   - Github Copilot Demo (30 min), the next step: Agents Demo (10 minutes)
5. **Report** back from **posterboard exercise** (Flavia) (2 min)

OPEN DISCUSSION LOGISTICS

1. We **split into groups** of 3-8 people (you can make your group).
2. We will pass out papers with the questions on them.
3. Each group will **report back a set of questions of their choosing** – there are four question sets. Prepare a 2 minute response for the report back.

STICKY DOTS EXERCISE AT TRB 2025



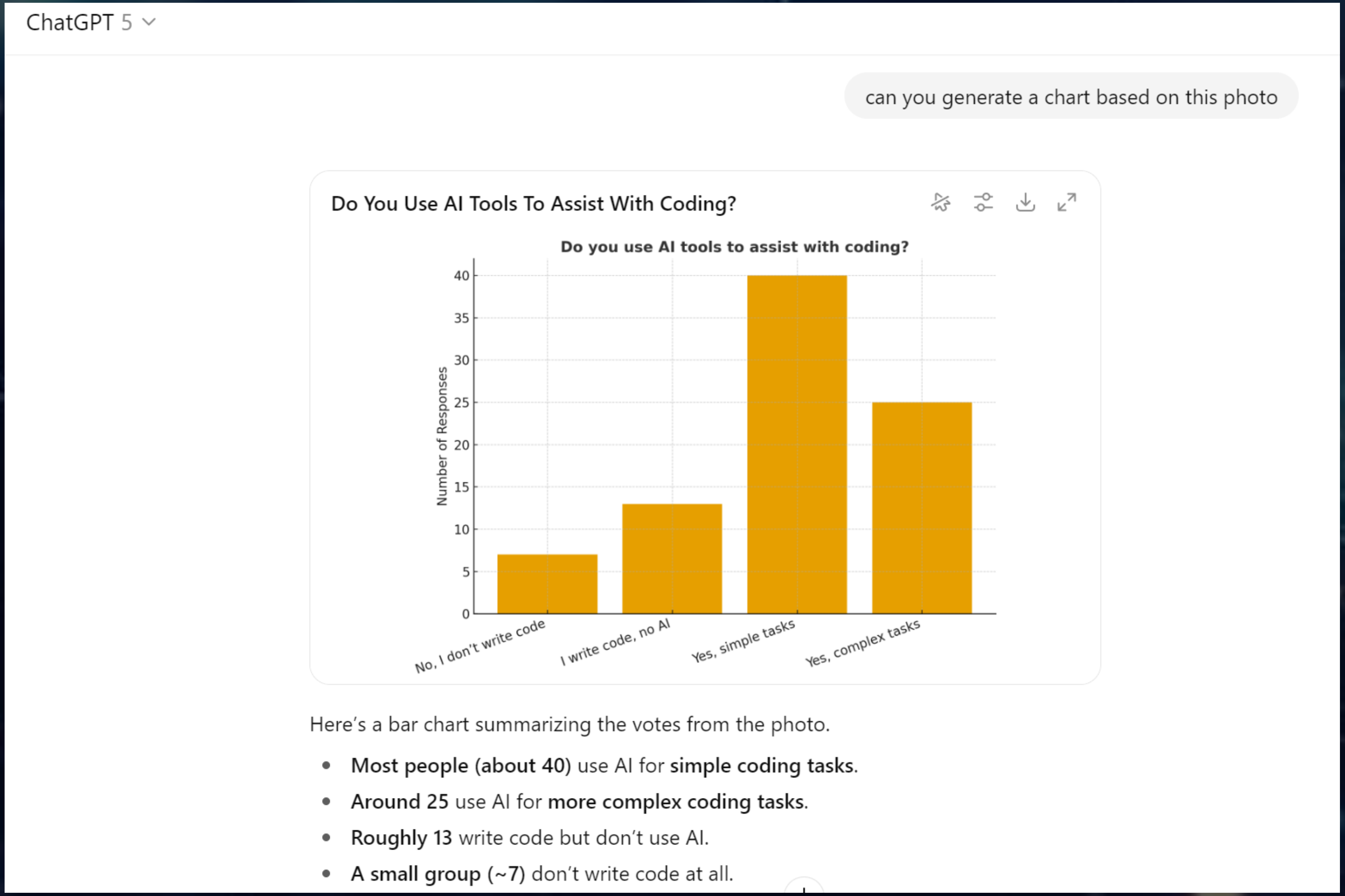
Do you use AI tools to assist with coding?

(1) No, I don't write code at all.

(2) No, I write code but I don't use AI

(3) Yes, for simple tasks e.g. Code snippets, syntax corrections

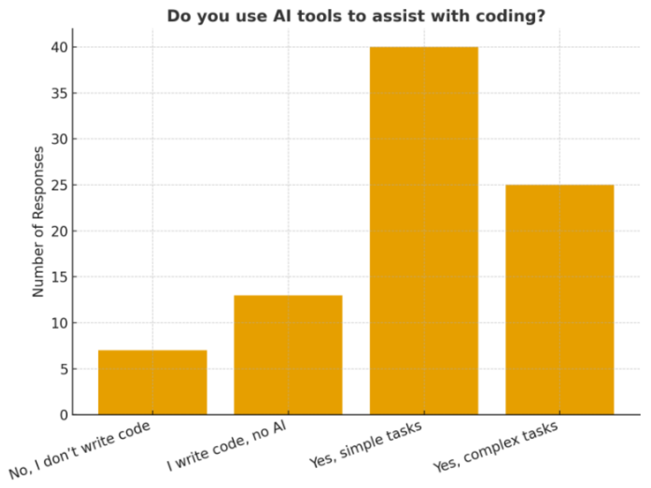
(4) Yes, for somewhat complex tasks e.g. debugging, code optimization



ChatGPT 5

can you generate a chart based on this photo

Do You Use AI Tools To Assist With Coding?



Here's a bar chart summarizing the votes from the photo.

- **Most people (about 40)** use AI for **simple coding tasks**.
- **Around 25** use AI for **more complex coding tasks**.
- **Roughly 13** write code but don't use AI.
- **A small group (~7)** don't write code at all.

TEAMWORK AND INSTITUTIONS

1. What institutional barriers have you encountered in trying to use AI tools in your programming work, e.g. budgets, policies? How have you been able to overcome them?

2. How can you work with a team of people and use AI tools together? What have you found in your team integration of AI?

3. Have you found good trainings for AI in coding that you can share?

4. How has your team been sharing their experiences and best practices for AI in coding? Do you have recommendations for teaching and sharing?

SPECIFIC AI TOOLS FOR PROGRAMMING

5. Which AI tools have you been using to do your programming work?

6. Do you have a recommendation for the best AI tools for our programming work? Why? List each tool you have used and its strengths and weaknesses?

HARD-WON KNOWLEDGE

7. On which of your programming tasks has AI tools been particularly useful and why? Which tasks has it failed on

8. Have you learned any useful tips and tricks for using AI for programming? What would you share with a coder who is new to using AI for programming?

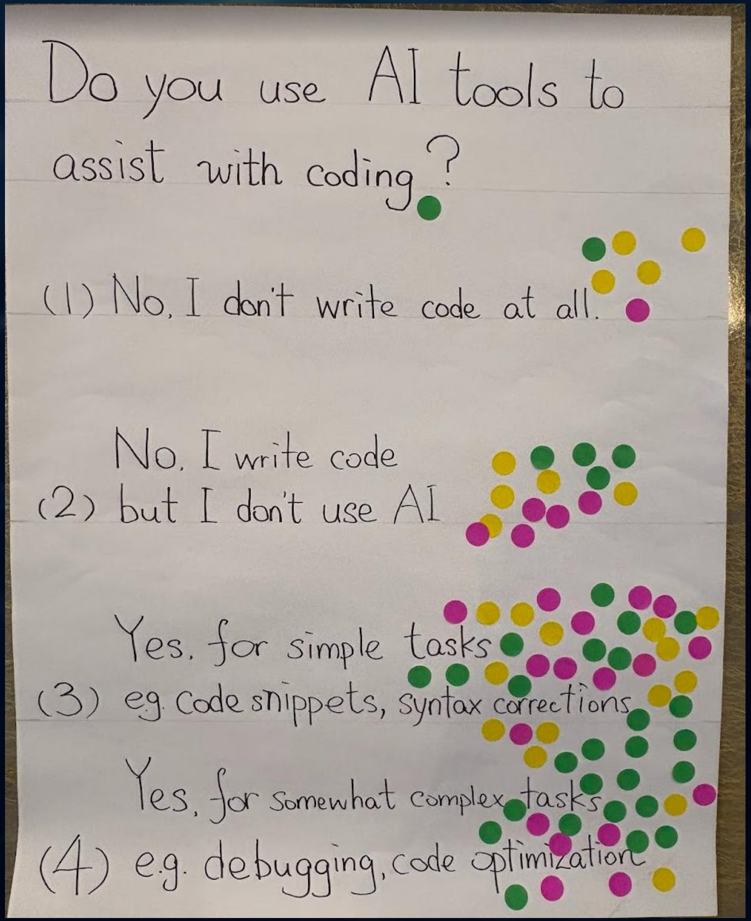
9. What bad experiences have you had in using AI for programming? What did you learn that you would like to share with the field?

THE DARK DOWNSIDES OF AI FOR PROGRAMMING

10. What have you noticed as the downsides resulting from your or your team’s use of AI? What can you or our field do to mitigate these downsides?

11. What do you fear are some negative outcomes that will result from greater use of AI in programming in our field?

STICKY DOTS EXERCISE AT TRB 2025

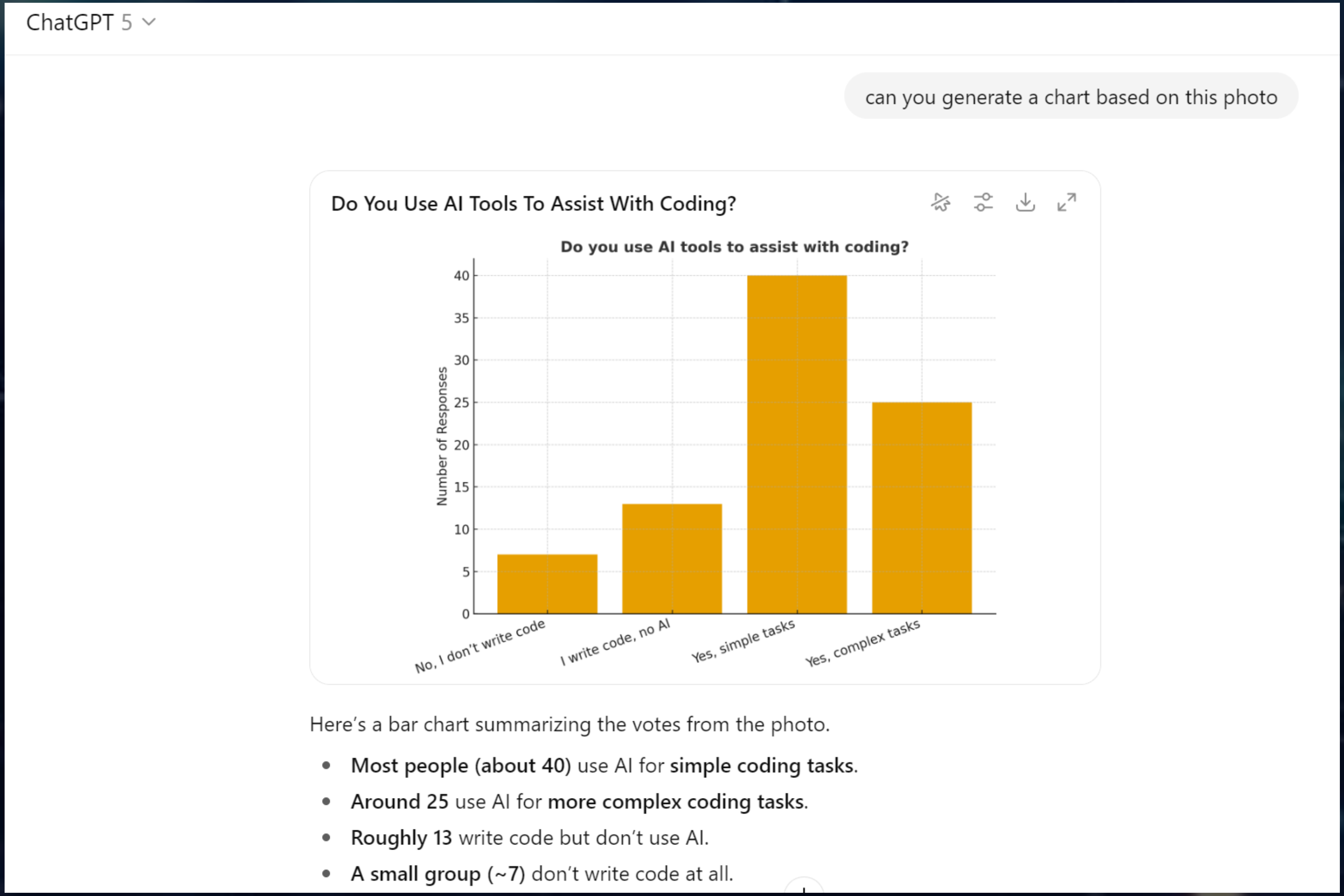

Do you use AI tools to assist with coding?

(1) No, I don't write code at all.

(2) No, I write code but I don't use AI

(3) Yes, for simple tasks e.g. code snippets, syntax corrections

(4) Yes, for somewhat complex tasks e.g. debugging, code optimization


ChatGPT 5

can you generate a chart based on this photo

Do You Use AI Tools To Assist With Coding?

Here's a bar chart summarizing the votes from the photo.

- **Most people (about 40)** use AI for **simple coding tasks**.
- **Around 25** use AI for **more complex coding tasks**.
- **Roughly 13** write code but don't use AI.
- **A small group** (~7) don't write code at all.

AI TOOLS + CODE ASSISTANCE

CODE COMPLETION v/s CODE GENERATION

	Code Completion	Code Generation
Goal	Helps you write code faster by suggesting next lines	Writes larger code blocks or scripts from scratch
How it works	Suggests next token/line based on context	Uses natural language to generate functional code
Typical Use	Filling in loops, arguments, boilerplate	"Write a function to group trips by corridor"
Tools	GitHub Copilot, IntelliSense, Tabnine	ChatGPT, Claude, Replit Ghostwriter
Scope	Local, one-line or block	Full function, script, or even a web app
Input	Code as you type	Prompts, comments, natural language

WHY USE AI FOR CODE GENERATION

Faster development

Reduce boilerplate code

Encourage exploration & iterations

Lowers the barriers for non-coders

AI TOOLS

	Use Case	Notes
GitHub Copilot	Code suggestions and completions	Embedded in VSCode, great for Python, SQL, etc.
Amazon Code Whisperer	Similar to Copilot	AWS integration, strong on cloud workflows
Tabnine	Lightweight code completion	Works across many editors, privacy-focused
ChatGPT Code Interpreter	Natural language to code	Great for data analysis, file transformations
OpenAI Functions / GPT API	Backend automation	Auto-generate routes, validation, parsing logic

LOW CODE NO CODE OPTIONS

	Use Case	Notes
Microsoft Power BI / Power Apps	Dashboards, app building	Configure data filters, maps, metrics
Alteryx	Data prep & modeling	Drag-and-drop workflows
Tableau Prep	Data cleaning & reshaping	Visual interface for data pipelines
Knime / Orange data mining	ML & analytics	Good for advanced users
JASP	Data Analysis & ML	Drag and drop workflows, beginner friendly

GITHUB COPILOT

GITHUB COPILOT ??

- An AI-powered coding assistant developed by GitHub and OpenAI
- Like autocomplete on steroids – suggests full lines, functions, or entire code files.
- Works inside VS Code, JetBrains, or GitHub Codespaces.
- Think of it as pair-programming + AI

FEATURES

	Scope of Changes	Interaction frequency	Developer Canvas
Completion	Next few lines	Hundreds of millisecond	VS Code (Editor)
Chats/Edits	Multifile Edits	Seconds	VS Code (Chat)
Agent Mode	Complete tasks	Minutes	VS Code (Chat)
Copilot coding agent	Entire issues	Tens of minutes	Github.com

GITHUB COPILOT IN WORKING..

- Powered by OpenAI Codex, a large language model trained on billions of lines of public code (GitHub, StackOverflow, docs)
- Copilot reads your content: comments, file names, and code around the cursor.
- It suggests code inline as you type – updated in real-time
- It learns from
  - Function names
  - Comments like "# load CSV and clean nulls"
  - Variable names
  - Previous files in your workspace

UNDER THE HOOD

LLMs DO these well

Code generation, completion and translation

Knowledge recall based on pre-training

Planning and problem solving

Pattern recognition

UNDER THE HOOD

LLMs DO NOT do these well

Real time
data access

Untrained
Knowledge

Specialized domain
expertise

Perfect
accuracy

UNDER THE HOOD

GitHub Copilot DO these well

Programming languages & common practices

Documentation and GitHub integrations

General Knowledge

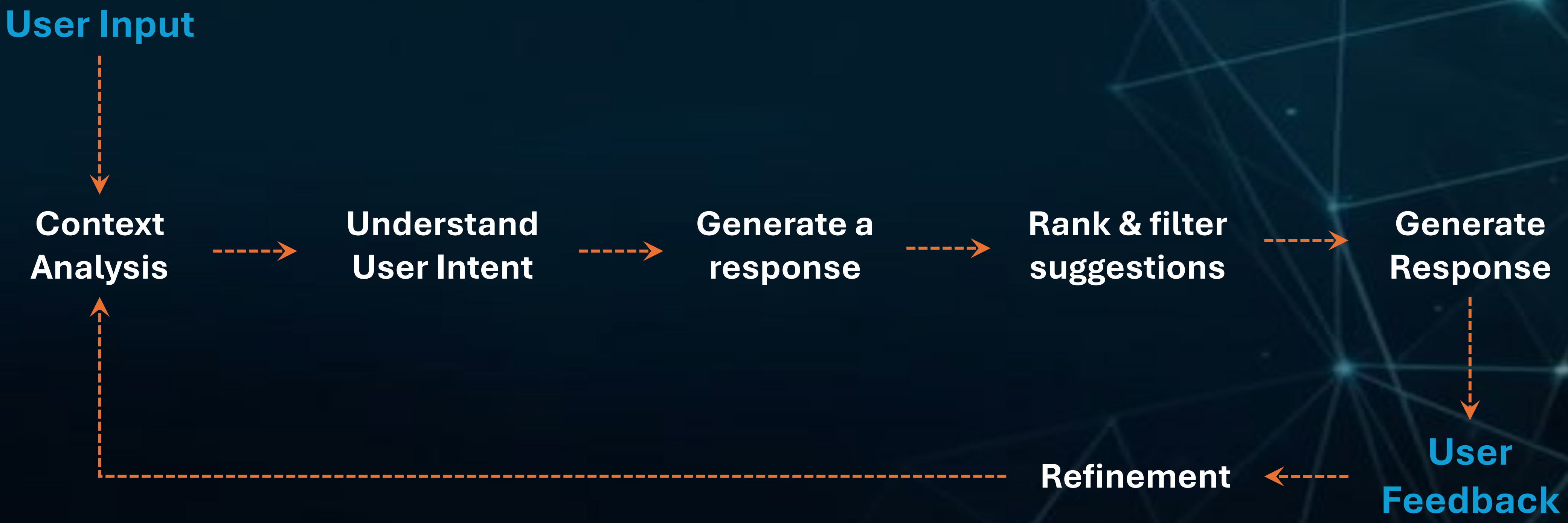
Code quality And safety

UNDER THE HOOD

GitHub copilot DO NOT do these well

- Access to private or proprietary code
- Full context of private GitHub repositories
- Original research or critical thinking
- Real time data or events

COPILOT RESPONSE CYCLE



WHY USE COPILOT?

- Faster Development
  - Cuts boilerplate time (e.g. reading/writing files, web scraping, APIs)
- Fewer Context Switches
  - Stay in the editor – no need to Google every syntax
- Learning While Coding
  - Great for junior devs or those learning new languages or libraries
- Supports Many languages
  - Python, JavaScript, SQL, R, HTML/CSS, C++, YAML, Markdown, and more
- Great for Data work
  - Quickly generate ETL scripts, model templates, SQL queries, etc.

BEST USE CASES OF COPILOT

Data Processing
Pandas/Numpy workflows

SQL Generation
Building queries from scratch

Code clean up
Auto-fixing legacy functions

Testing
Unit test generation

LIMITATIONS & CONSIDERATIONS

- **Not always Right**
  - Copilot may suggest insecure or non-optimal code
- **Be Mindful of Sensitive Data**
  - Don’t accept suggestions that might include leaked code/data
- **You’re still the Expert**
  - Always review, test, and validate its output

HAVE FUN WITH GITHUB COPILOT

ADDITIONAL TIPS

- Use descriptive function names and comments
- Think in "intent" (Copilot responds better to what you're trying to do)
- Accept, edit, or ignore suggestions as needed.
- Combine with GitHub Copilot Chat (for Q&A and debugging)