My Journey Building an R-Generating AI Coding Agent: Lessons from a Statistical Programmer

Ippei Akiya, Clindroid

Accuracy came not from a smarter model, but from giving the agent clear procedures, boundaries, and output contracts.

1 Background

AI agents are rapidly changing software development. But adoption is still early in clinical statistical programming.

What statistical programming requires:

Data confidentiality	Trial data must not leak outside. Use, transfer, and store it securely.
Regulatory and data standard compliance	Conformance to data standards is strongly required.
CSV activities	The computing environment and libraries must be validated.

Can we bring the same paradigm into a regulated, accuracy-critical field?

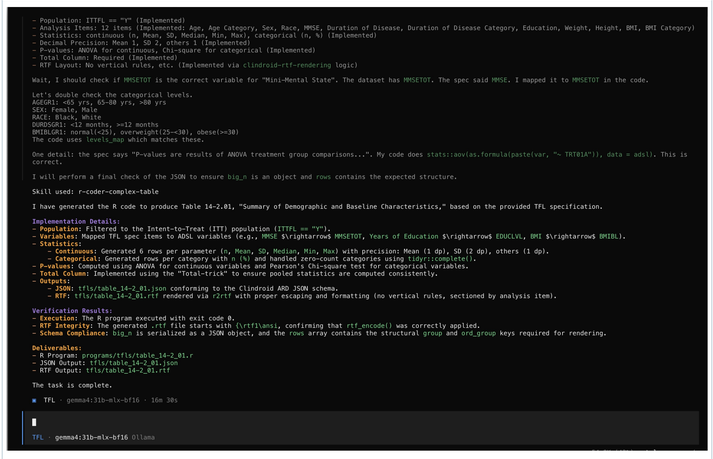
2 Downloading Clindroid Buddy

We decided to build Clindroid Buddy, a coding AI agent specialized for statistical programming.

We modified OpenCode, an open-source coding AI agent, to specialize it for statistical programming.

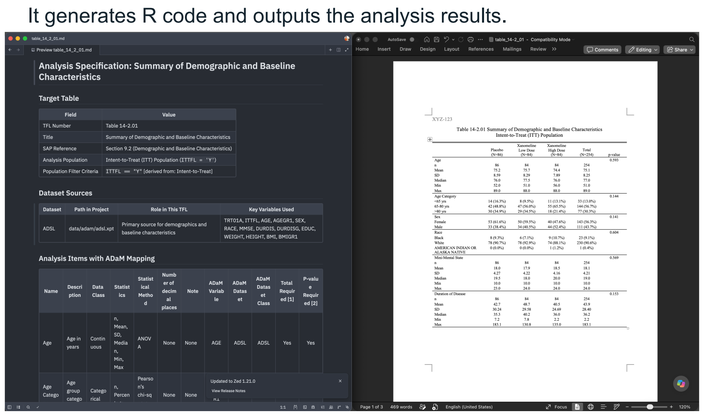
https://github.com/Clindroid/Clindroid-Buddy

3 Clindroid Buddy User Interface



4 TFL Spec and Output

Created autonomously by the AI agent
It generates R code and outputs the analysis results.



5 Feature List

Feature	Dev. status
SDTM creation (incl. spec, define.xml, and SDRG)	Not implemented
ADaM creation (incl. spec, define.xml, and ADRG)	Not implemented
TFL creation (incl. spec)	Partly implemented
RAG (reference to the Model Specification and Implementation Guide of SDTM, ADaM, etc.)	Implemented
CDISC Terminology download	Implemented

* SDTM, ADaM, and TFL are produced by generating R code and then running it.

The generated R code runs automatically, and the tool repeats fixes until no errors remain.

6 Not Generated as Intended

Problem
The agent could not read the SAP and shells and generate R code for tables directly (using the Gemma 4 31B model).

Symptom
Errors in the R code could not be fixed, or the analysis results were wrong.

Cause
At each fix, the AI often changed correct parts of the code along with the wrong parts.

Solution
Accuracy improved when the AI first wrote a spec and then generated R code focused on that spec.

7 Unrequested File Read/Write

Problem
The agent read files and data it was not told to read, and wrote files to directories other than the specified one.

Symptom
It wrote intermediate datasets and program files into the ADaM directory. It also read and used the wrong dataset.

Cause
Too little instruction, plus cases where the AI did not follow it.

Solution
Use Agent SKILL, and limit access with permission settings.

8 Hard-to-Read R Code

Problem
It generates R code that is hard for humans to understand. The results are often wrong.

Symptom
It does not use join or bind_rows well. There is no established approach for shaping summary statistics of several variables into the final table.

Cause
Not enough knowledge about outputting several analysis results as one table in R, or not enough reasoning ability in the LLM.

Solution
Write the strategy for building the final analysis table in the Agent Skill.

9 Agent Mode

Modes: four custom modes added — SDTM / ADaM / TFL / Review.
Each agent is limited to certain directories for read/write.

Agent	data/ sdtm	data/ adam	specs/ tfls	prog/ tfls
SDTM	write	deny	deny	deny
ADaM	read	write	deny	deny
TFL	read	read	write	write
Review	read	read	read	read

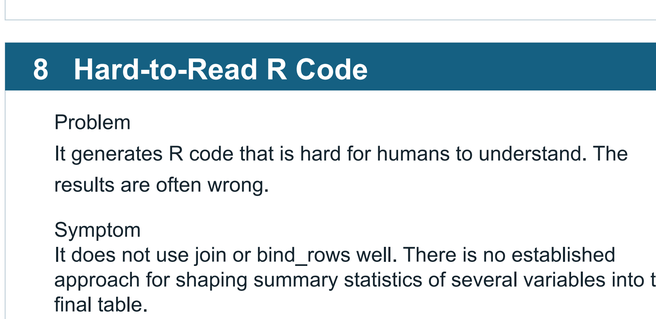
In addition, each agent is assigned its own Agent Skills and tools.
As a result, the agent no longer hesitates over which directory to use or which skill and tool to choose. This raises accuracy and reduces wasted tokens from repeated searching.

10 Prompts and SKILLs

How agent modes relate to AGENTs.md, System Prompt, and SKILL.

Agent	AGENTs.md	System Prompt	SKILL
SDTM	Common Defines directory structure and versions	Set differently for each agent	Each agent can use a different set of SKILLs
ADaM			
TFL			
Review			

11 Steps to TFL Output



12 Challenges in RTF Output



Separate statistics from formatting → diff, review, and audit
A leaner schema is followed more reliably

Fine control of RTF layout with r2rtf was hard: perhaps because the r2rtf help is too brief for the LLM to understand, it repeated trial and error or could not produce some layouts at all.
Detailed instructions in a SKILL made it work, but a mechanism for the agent to research and solve problems on its own still has room for improvement.

13 Reasoning Ability and Cost

We are tuning the setup so that Gemma 4 31B gives good results.
A stronger LLM might generate R code more efficiently, but inference cost becomes a problem.

Model	Input tokens	Output tokens
Gemma 4 31B	**0.14 USD**	**0.40 USD**
Claude Sonnet 5	**2.00 USD**	**10.00 USD**
Claude Opus 5	**5.00 USD**	**25.00 USD**

* Input and output token costs are per 1 million tokens. Compared using Amazon Bedrock (US East region) prices.
Compared with the Claude models used in software development, the gap is tens of times larger.

14 Thanks to the OpenCode Developers

Our sincere thanks to the OpenCode developers and community for making this work possible.
Its agent modes, Agent Skills, and permission model gave us the foundation to specialize the tool for statistical programming — we did not have to build an agent from scratch.

OpenCode GitHub repository:
https://github.com/anomalyco/opencode