INFFER Project: AI-Powered R Code Feedback

A Scalable Learning Assistant for R Coding and Data Analytics

Ulrich Matter & WDDA Team

December 15, 2025

Bern University of Applied Sciences (BFH)

Supported by BFH E-Learning funding program (virtuelle-akademie.ch)

AI tools used: Claude Code (Sonnet 4.5, Opus 4.5) for app framework, testing, deployment, bug fixing, CSS styling, Beamer setup

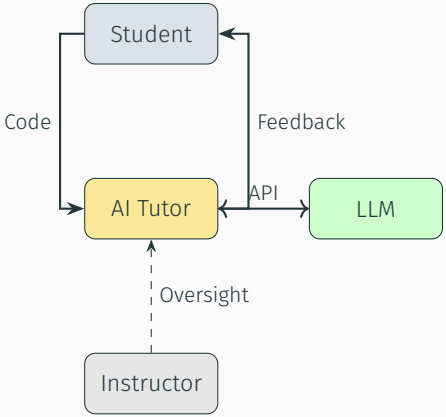
The Challenge: Scaling R Coding Education

Teaching R coding at scale presents unique challenges:

- Students need immediate, personalized feedback
- Instructors face high grading workloads
- Traditional office hours don't scale
- Students hesitate to ask "basic" questions

Our approach:

Leverage large language models (LLMs) as on-demand tutoring assistants.



Conceptual Background: AI in Education

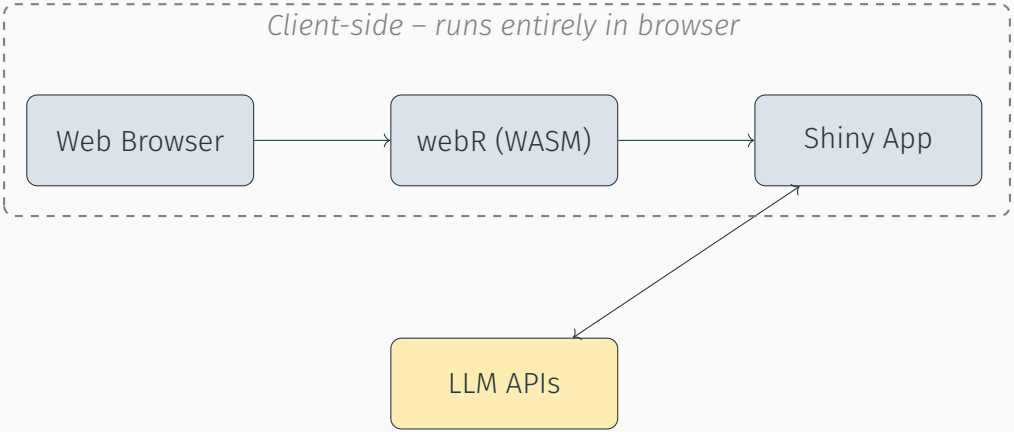
Formative feedback is critical for learning R coding:

- Timely feedback improves learning outcomes (Hattie & Timperley, 2007)
- Self-regulated learning requires actionable guidance
- Positive framing reduces anxiety and encourages experimentation

Why LLMs for code feedback?

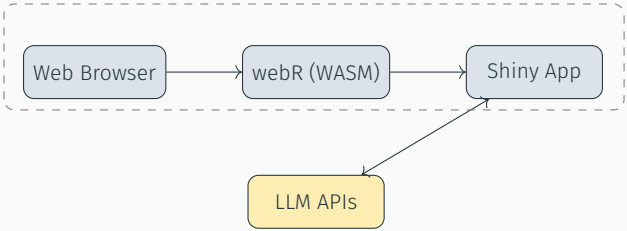
Capability	Traditional Tools	LLM-Based
Syntax checking	✓	✓
Unit tests / correctness	✓	✓
Style suggestions	Rule-based	Contextual
Conceptual explanations	Pre-written	Adaptive
Personalized tone	–	✓
Natural language interaction	–	✓

Architecture: Serverless & Privacy-First



OpenRouter / OpenAI / Anthropic

Architecture: Key Design Principles



Key design principles:

- **Privacy by design**: No student code stored on any server
- **"Zero" infrastructure**: Static hosting via GitHub Pages
- **BYOK model**: Students use their own API keys (which we hand out)
- **Multi-provider**: Flexible model selection (via OpenRouter)

Three-Layer Feedback System

Layer 1: Syntax Check – Immediate R parsing validation <1 sec

Layer 2: Rule-Based – Pattern matching (works offline) <1 sec

Layer 3: AI Feedback – LLM analysis with pedagogical prompting 3–10 sec

AI feedback is pedagogically prompted to:

- Start with positive observations
- Frame suggestions as learning opportunities

Features & Intended Use

Key Features:

- Multi-language UI (EN/DE/FR)
- Screenshot paste support
- Configurable AI models
- Free tier options available
- Mobile-responsive design

Recommended Models:

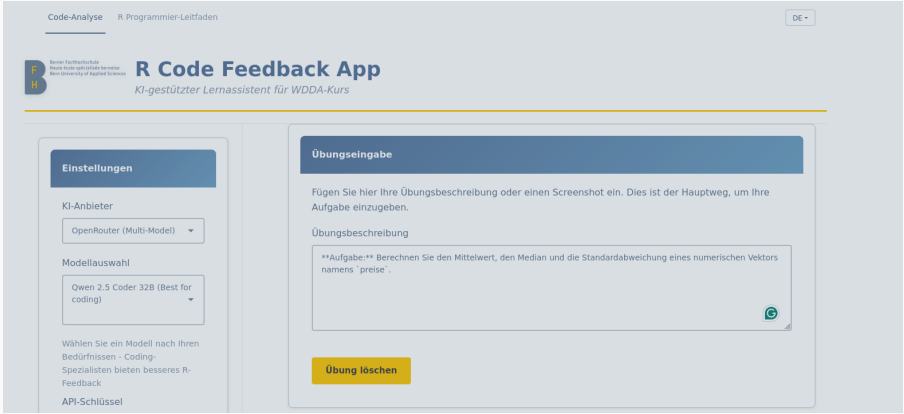
- Qwen 2.5 Coder – Best for code
- DeepSeek V3 – Best value
- Gemini Flash – Free tier

Intended Use Cases:

1. **Self-study**: Students practice independently
2. **Homework support**: Get unstuck without waiting
3. **Exam prep**: Test understanding with instant feedback
4. **Flipped classroom**: Pre-class coding exercises

Not intended to replace instructor grading.

Enjoy the Show



Summary

What we built:

- AI-powered R code feedback tool
- Runs entirely in browser (no server)
- Privacy-preserving design
- Multi-model, multi-language support

Technology stack:

- R + Shiny + Shinylive
- webR (WebAssembly)
- OpenRouter / OpenAI / Anthropic APIs

Try it out

Deployed at:

`https://umatter.github.io/inffer_feedback_app`

Source code:

`https://github.com/umatter/inffer_feedback_app`

Scan to open app

Questions?