

Agentic Programming with Local Models

A Walk-Through with Examples

ILLINOIS

Dirk Eddelbuettel

Public Lecture, 18 Nov 2026

ILLINOIS

Overview

Scope

ILLINOIS

This talk will

- undoubtedly reflect my personal view and preferences
- in no way attempt to cover *all* possible tools / approaches
- use R as a running example *but the underlying machinery is portable*
- focus on *local* models
- demo on *middle-of-the road basic retail hardware* (i.e. my NVidia RTX 5060)
- take a Linux perspective (which in my case is Ubuntu, now at 26.04)
  - but if someone using macOS reads and wants to adapt / extend, great!
  - send me with PRs or issues to the Github repo

So Why Local Models? (1 of 3)

ILLINOIS

Some key reasons:

- No per-token billing, no running meter, free to experiment
- No subscription, no lock-in
- No ‘you ran out of your tokens’
- No intellectual property leakage
- No code leakage, no data leakage, no spying on activity
- Nobody else gets your data or code

So Why Local Models? (2 of 3)

ILLINOIS

Some key reasons (cont.):

- Supports old-school autonomous and off-line computing, say
  - laptop without wifi
  - remote area without broadband
- More control:
  - your machine, your upgrades, your versions
  - whereas cloud services ‘can disappear / change’
- Interface fully compatible with different hosting

So Why Local Models? (3 of 3)

ILLINOIS

Some key reasons (cont.):

- Mix and match is an interesting option
  - do first tasks with a (simpler, cheaper) local model
  - then refine with a frontier model
- Basic 80/20 rule applies again
  - for simpler talks, use simple(r) models
  - why pay an external model for a pedestrian task?

ILLINOIS

Local Models

What is it?

ILLINOIS

Basic understanding of local models:

- In essence, ensuring agent inference is executed locally
- Can be a ‘Yuge’:
  - Salvatore Sanfilippo (aka @antirez) writing ds4
  - biggest setup is ‘two 512gb mac mini’ streaming for 1tb ram
  - nice to see an open source, but few have that type of equipment
- But standard ‘retail’ setup can work too
- We will focus on such a ‘retail’ setup here
- Note though that ‘local’ can also be a 1tb ram server on a network

Our Setup

ILLINOIS

Concrete use case:

- A decent-ish machine from last year
  - NVidia Geforce RTX 5060 with 16gb VRAM, plenty cores (32) and ram (64gb)
  - Running standard Ubuntu 26.04
- The key constraint is the 16gb VRAM.
  - *In theory* different approaches (e.g. 4 bit quantization) should help
  - *In practice* this did not (yet ?) work for me.
- So (mostly) sticking with two 'open weight' models that fit in the 16gb.
  - `qwen3.5` (by Alibaba) and `gemma4` (by Google Deepmind)

Software

ILLINOIS

Preference for open source:

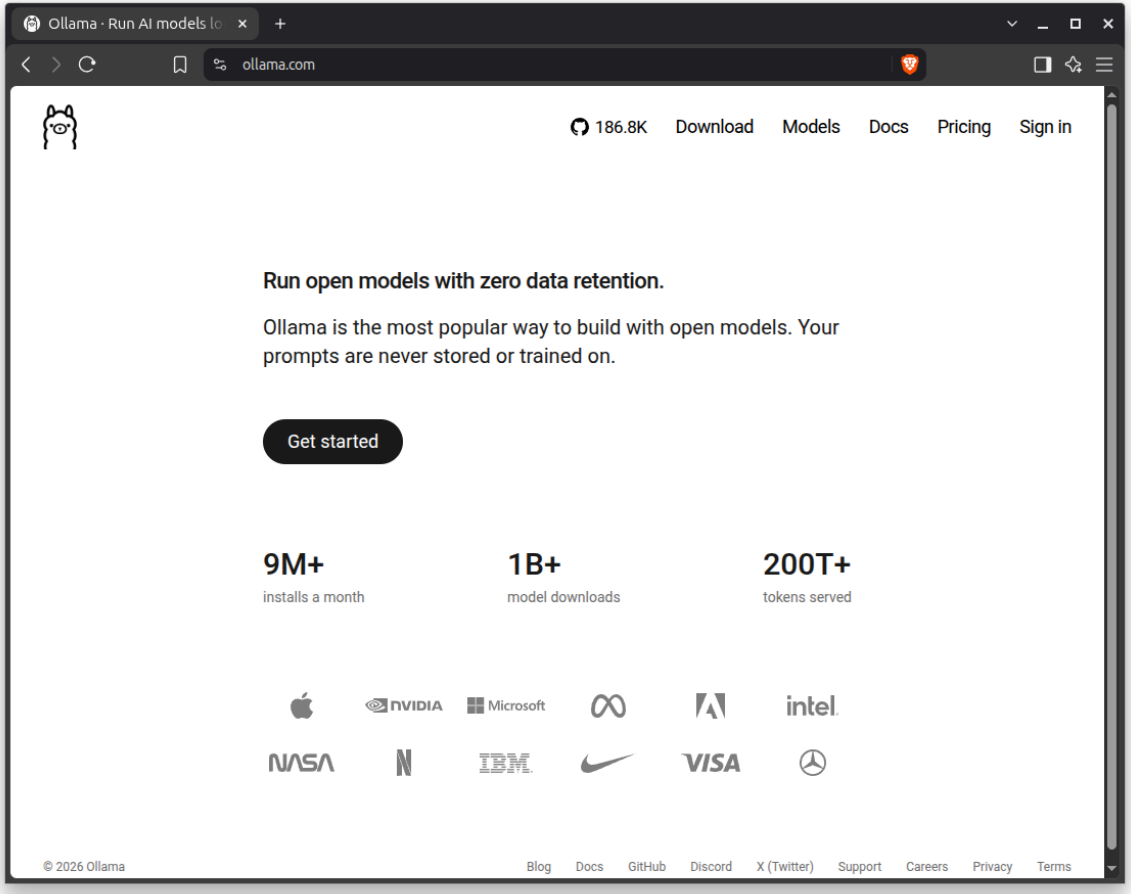
- Key here is llama.cpp (github) (wikipedia)
- Open source, C++ based, widely deployed in at least
  - ollama (more on this later)
  - LMstudio (which we looked into, and is promising too)
  - several R projects on CRAN which wrap or include llama.cpp:
    - llamaR (CRAN page)
    - localLLM (CRAN page)
    - edgemodelr (CRAN page)
    - relm (CRAN page)

ILLINOIS

Ollama

Ollama Website

ILLINOIS



About Ollama

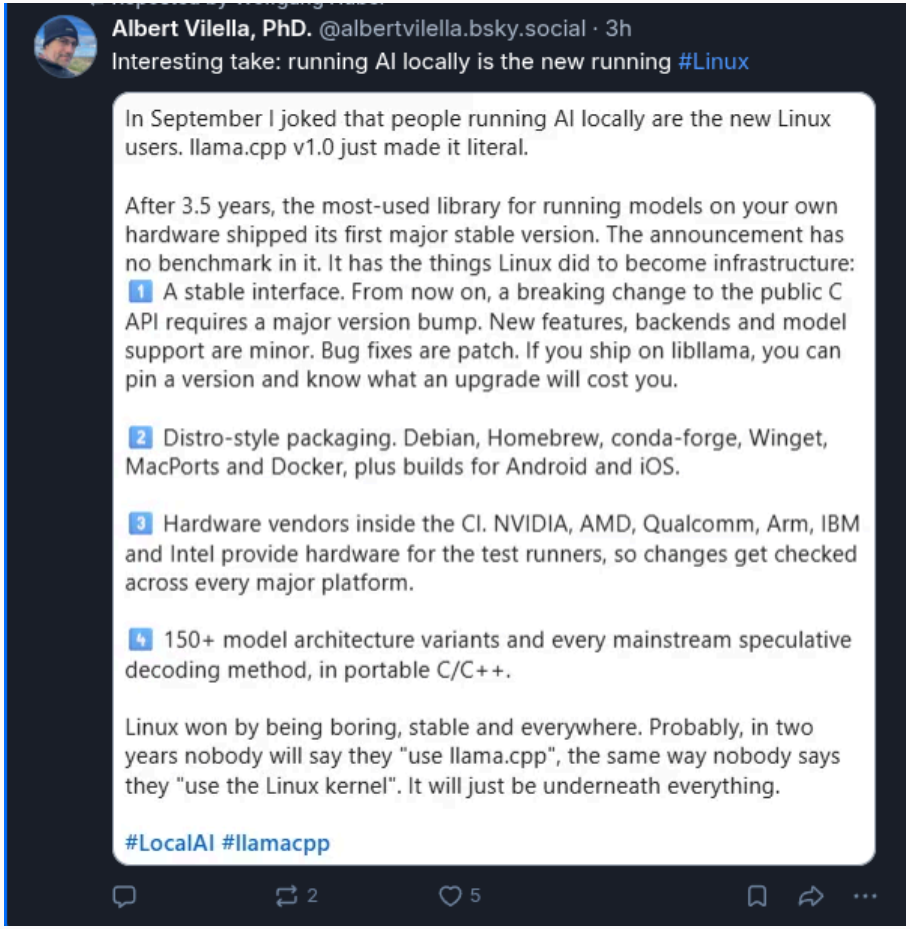
ILLINOIS

Key Features

- Provides the open-source `ollama` package
- Serves model weights for download and use with `ollama`
- Runs a cloud-service too to monetize
- Popular choice; more at https://en.wikipedia.org/wiki/Ollama
  - but note the criticism: really dependent on llama.cpp
- For my use case, *easy* as I can rely on a `.deb` package which
  - thanks to https://packages.lingfish.net/apt/ auto-updates
  - is fabulous as I get a new Ubuntu 26.04 binary every few days

Aside: llama.cpp 1.0 Praise

ILLINOIS



Albert Vilella, PhD. @albertvilella.bsky.social · 3h

Interesting take: running AI locally is the new running #Linux

In September I joked that people running AI locally are the new Linux users. llama.cpp v1.0 just made it literal.

After 3.5 years, the most-used library for running models on your own hardware shipped its first major stable version. The announcement has no benchmark in it. It has the things Linux did to become infrastructure:

1️⃣ A stable interface. From now on, a breaking change to the public C API requires a major version bump. New features, backends and model support are minor. Bug fixes are patch. If you ship on libllama, you can pin a version and know what an upgrade will cost you.

2️⃣ Distro-style packaging. Debian, Homebrew, conda-forge, Winget, MacPorts and Docker, plus builds for Android and iOS.

3️⃣ Hardware vendors inside the CI. NVIDIA, AMD, Qualcomm, Arm, IBM and Intel provide hardware for the test runners, so changes get checked across every major platform.

4️⃣ 150+ model architecture variants and every mainstream speculative decoding method, in portable C/C++.

Linux won by being boring, stable and everywhere. Probably, in two years nobody will say they "use llama.cpp", the same way nobody says they "use the Linux kernel". It will just be underneath everything.

#LocalAI #llamacpp

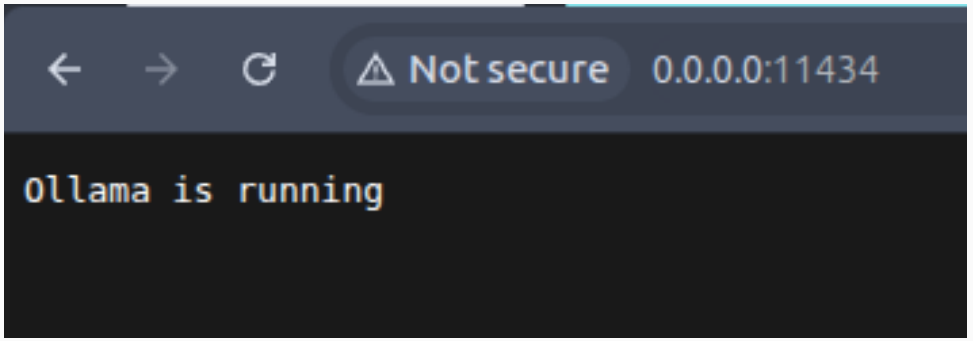
2 5

Deploying Ollama

ILLINOIS

Ollama Deployment

- By default it runs on port 11434
- The URL is typically one of these
  - http://localhost:11434
  - http://127.0.0.1:11434



- We prefer to use 0.0.0.0 instead as this
  - is open to the local network so another machine can access it
  - requires a small config setting (see below)
- Provides standard API first proposed by OpenAI via http://0.0.0.0:11434/v1

Deploying Ollama (cont.)

ILLINOIS

Ollama Setup

- Run `sudo systemctl edit ollama` and add these lines

```
[Service]
Environment="OLLAMA_HOST=0.0.0.0"
Environment="OLLAMA_KEEP_ALIVE=-1"
Environment="OLLAMA_CONTEXT_LENGTH=32768"
```

- Followed by

```
sudo systemctl daemon-reload
sudo systemctl restart ollama
```

- This ensures network use, longer sessions and an increased context window.
- It updates a file in `/etc/systemd/system/ollama.service.d/`.

Running Ollama

ILLINOIS

Basic commands

- `ollama serve` to start the server (*not needed in my case*)
- `ollama list` shows locally available models
- `ollama pull model` downloads named model
- `ollama run model` launches a named model
- `ollama rm model` removes a named model
- `ollama ps` shows running model(s)
- `ollama create new-model -f Modelfile` (see below)

Running Ollama (cont.)

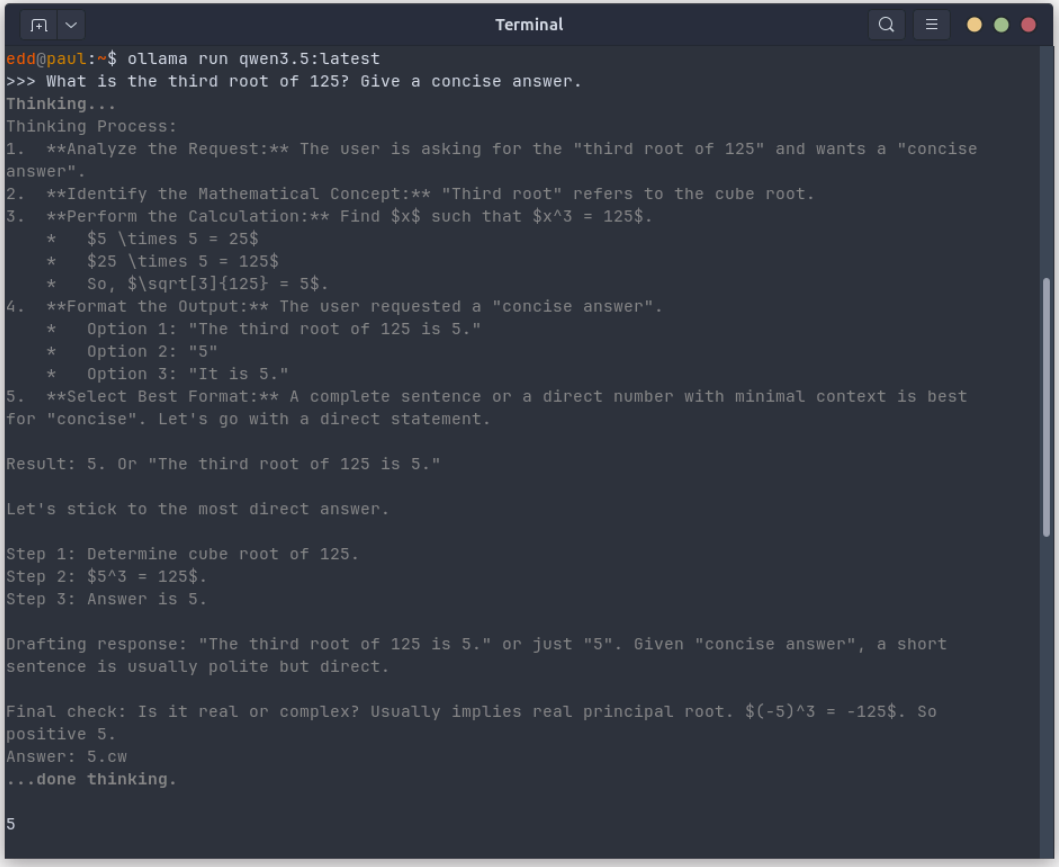
ILLINOIS

With Ollama running we can

- run it in the shell as a bare bones chat
- set up local tools such as `opencode` (next section) to deploy it
- use agents such as `pi` (which we used via a Docker container)
- use it from RStudio as a chat interface (but not for ‘posit assistant’)
- use it from Positron as a chat interface (but not for ‘posit assistant’)
- use it from Open WebUI in a browser at http://localhost:8080/
- connect to it from editors (I use Emacs, I am sure vim and vscode do too)

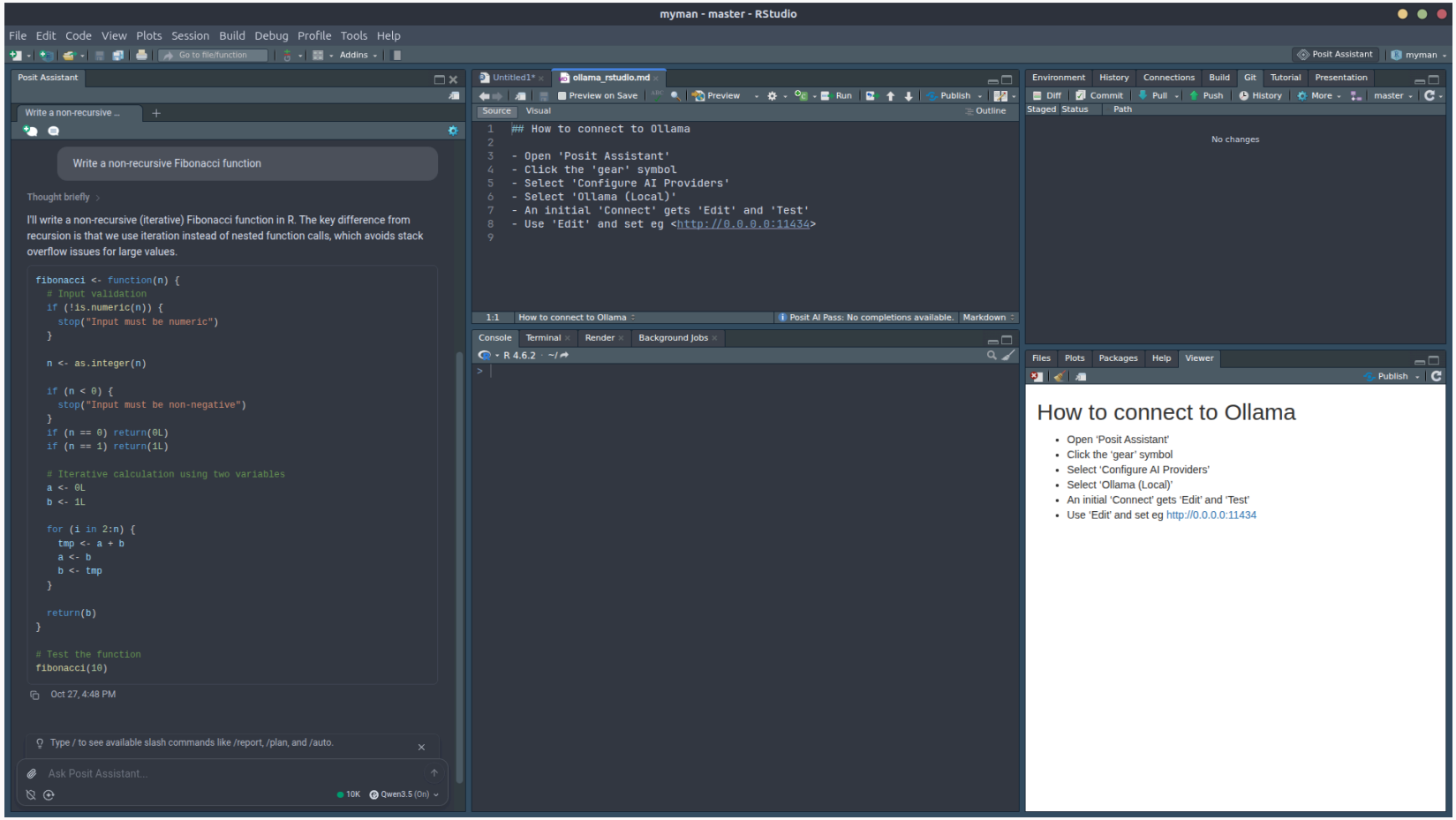
Ollama on the command-line

ILLINOIS



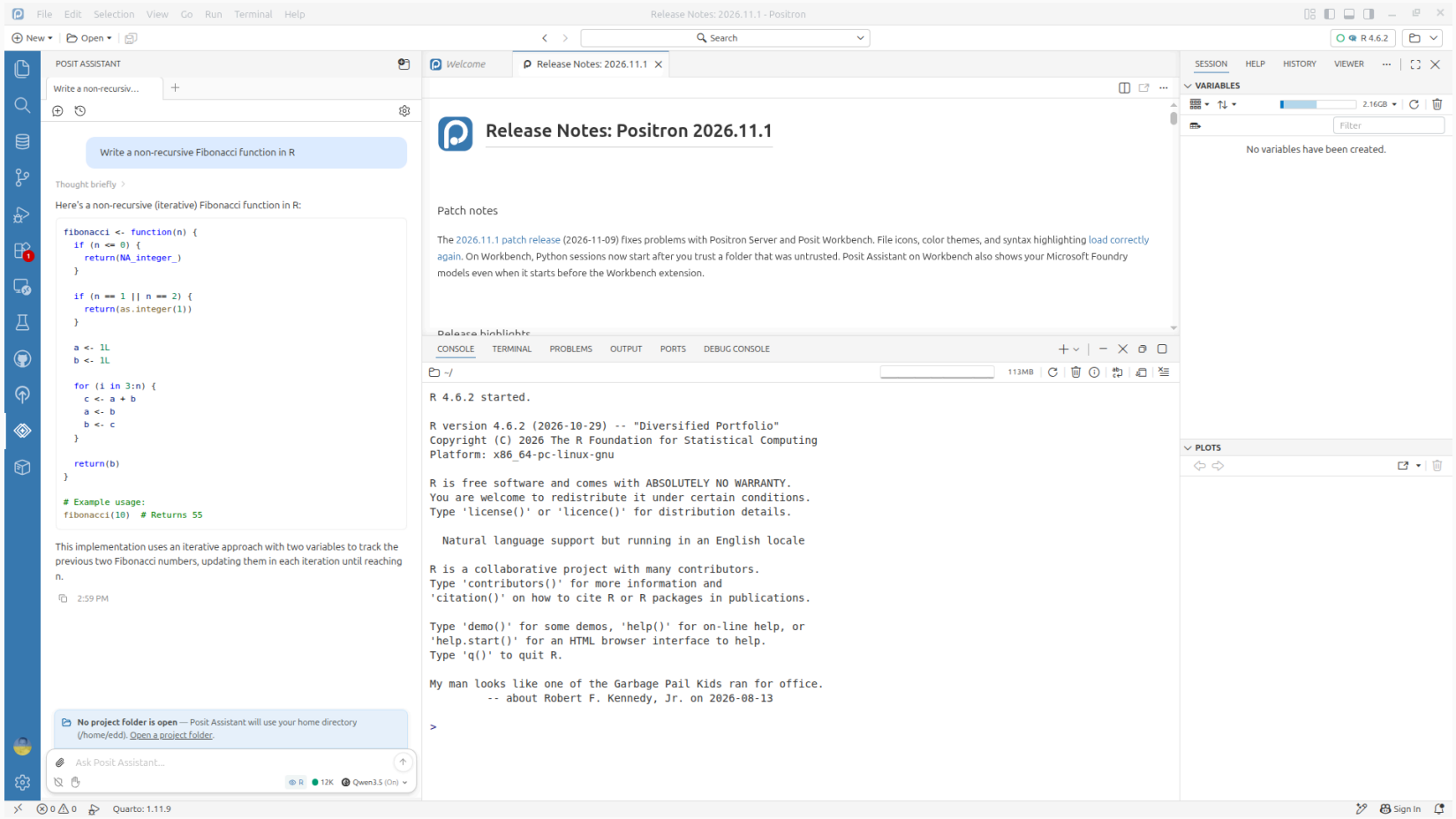
Ollama in RStudio

ILLINOIS



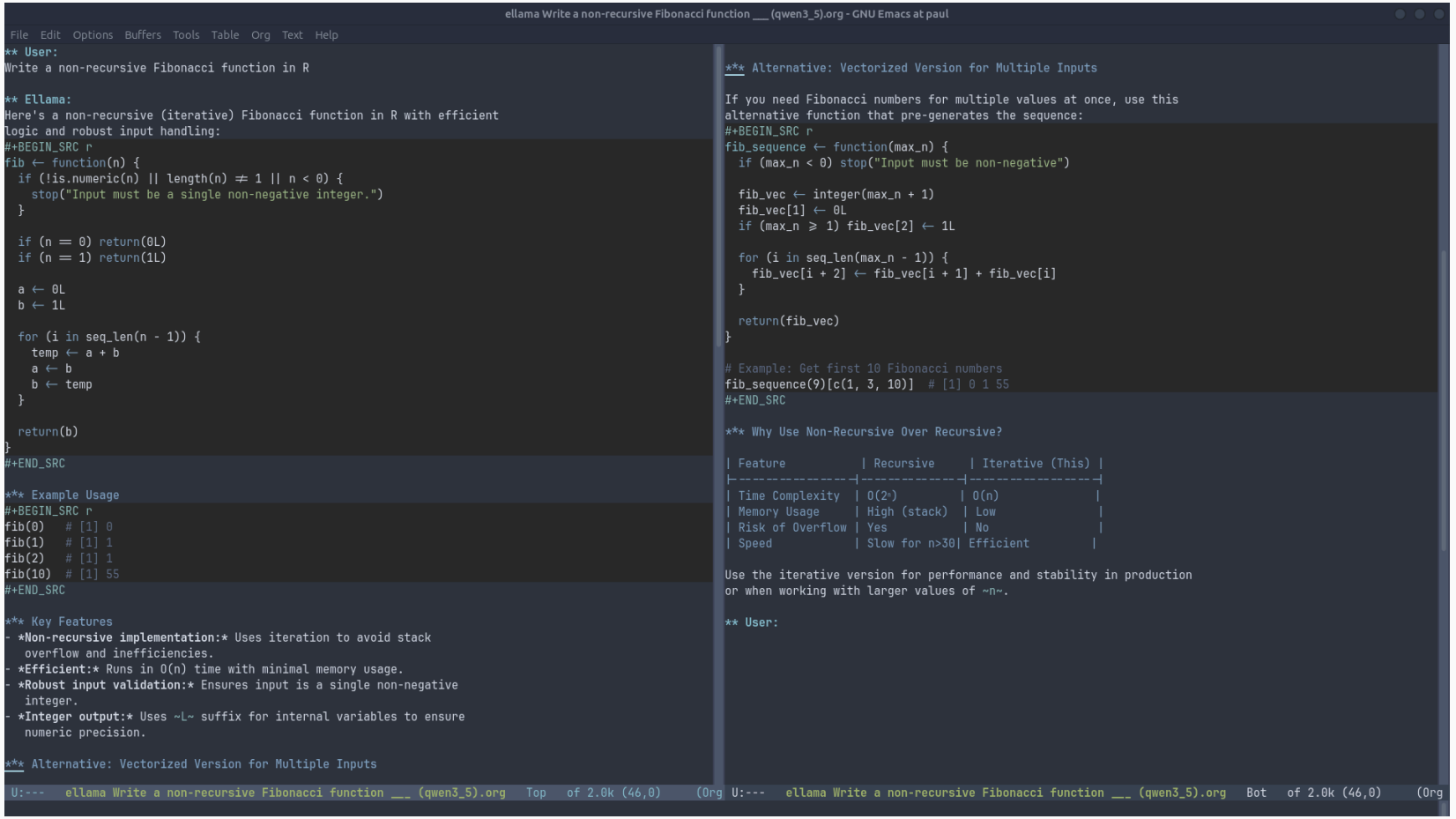
Ollama in Positron

ILLINOIS



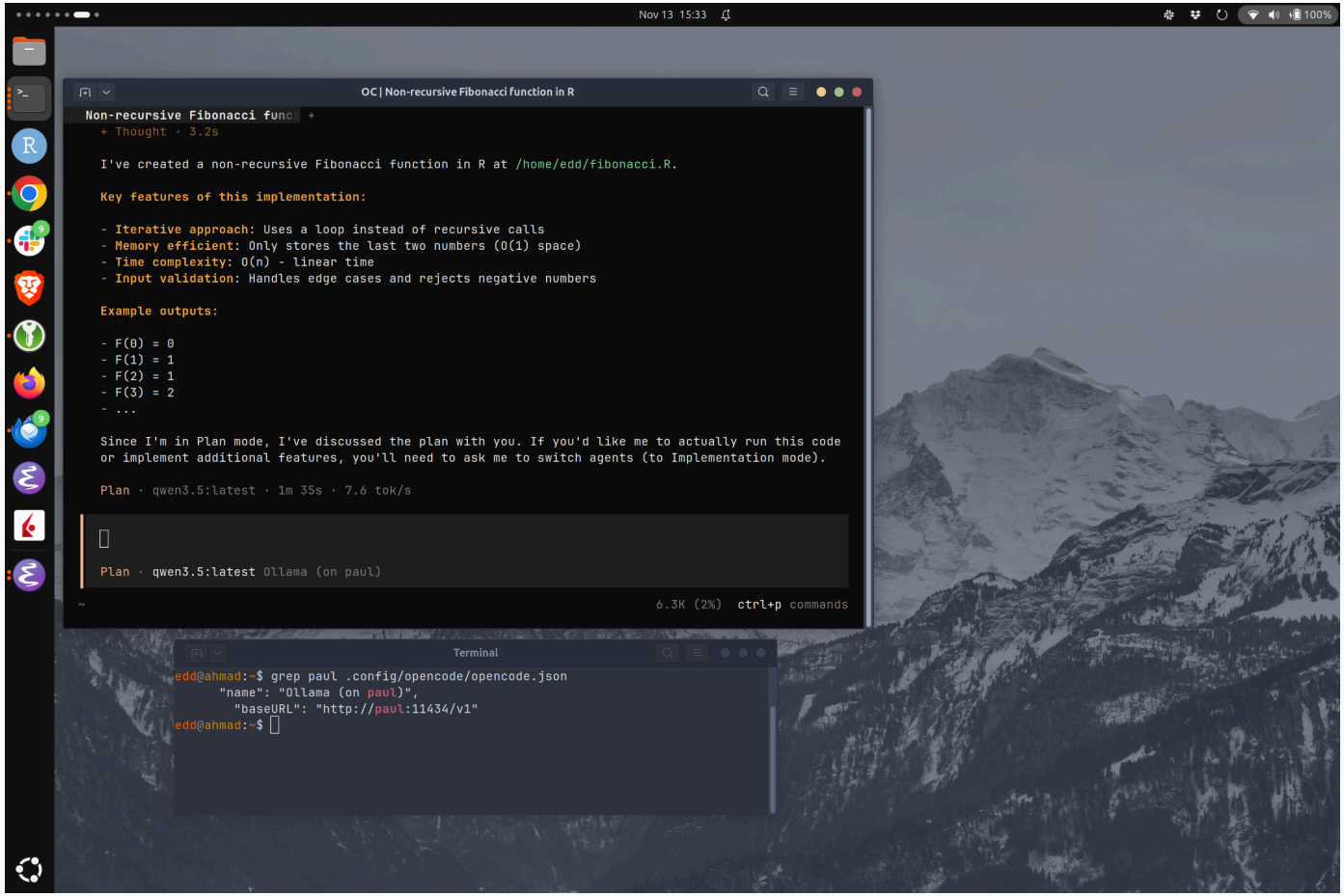
Ollama in Emacs with Ellama

ILLINOIS



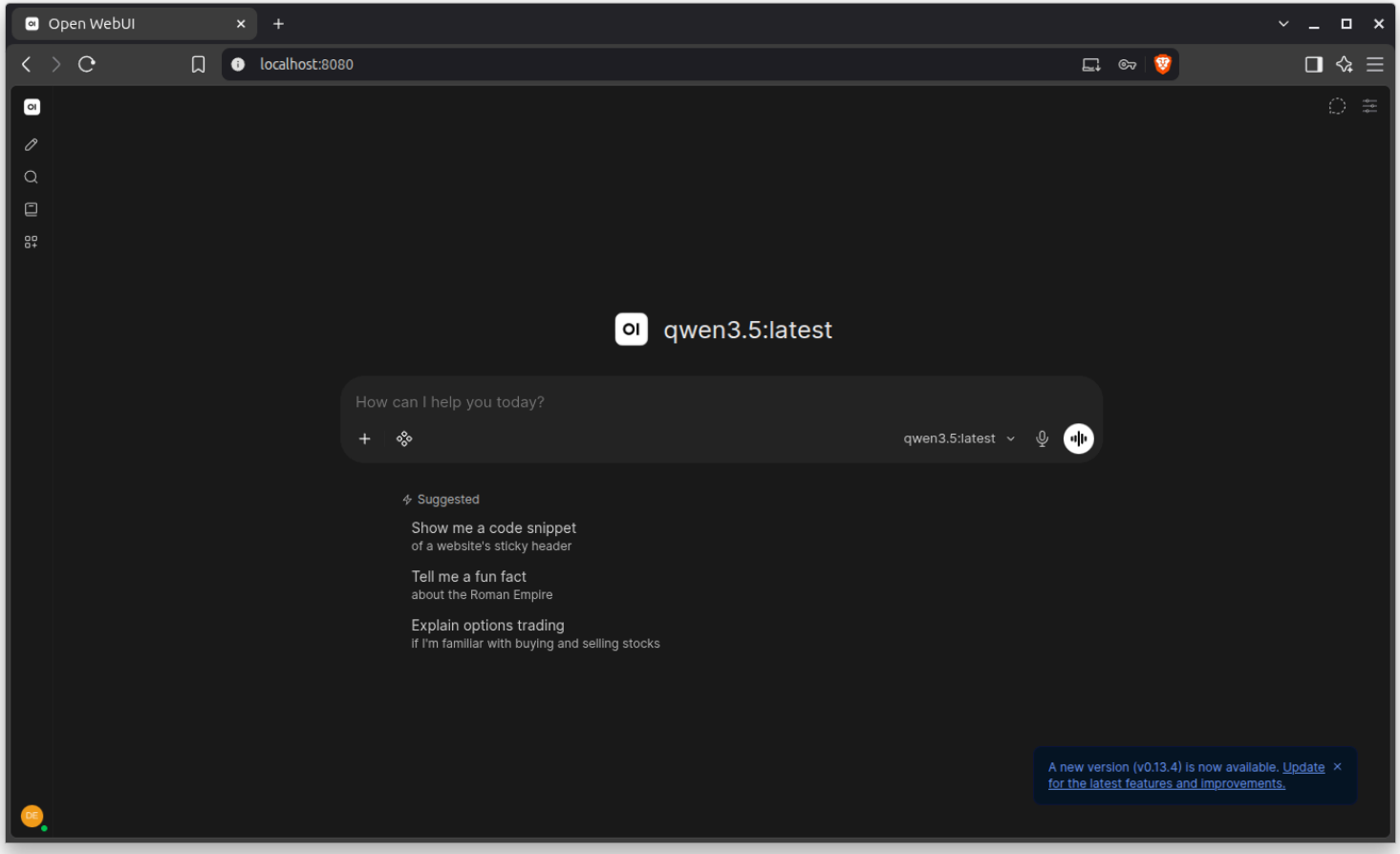
Ollama from Laptop to Server

ILLINOIS



Ollama via Open WebUI locally

ILLINOIS



Local Modelfiles

ILLINOIS

Modifying and Adapting

Using a `Modelfile` such as

```
FROM gemma4:latest
PARAMETER num_ctx 131072
```

we can create a local variant of a downloaded model via *e.g.*

```
ollama create gemma4-local -f ./Modelfile
```

which uses the instructions in `Modelfile` to create the named model.

This also works for downloads from HuggingFace, a popular model repository.

Attempts at ‘squeezing’ larger models to fit did not succeed though.

Ollama Summary

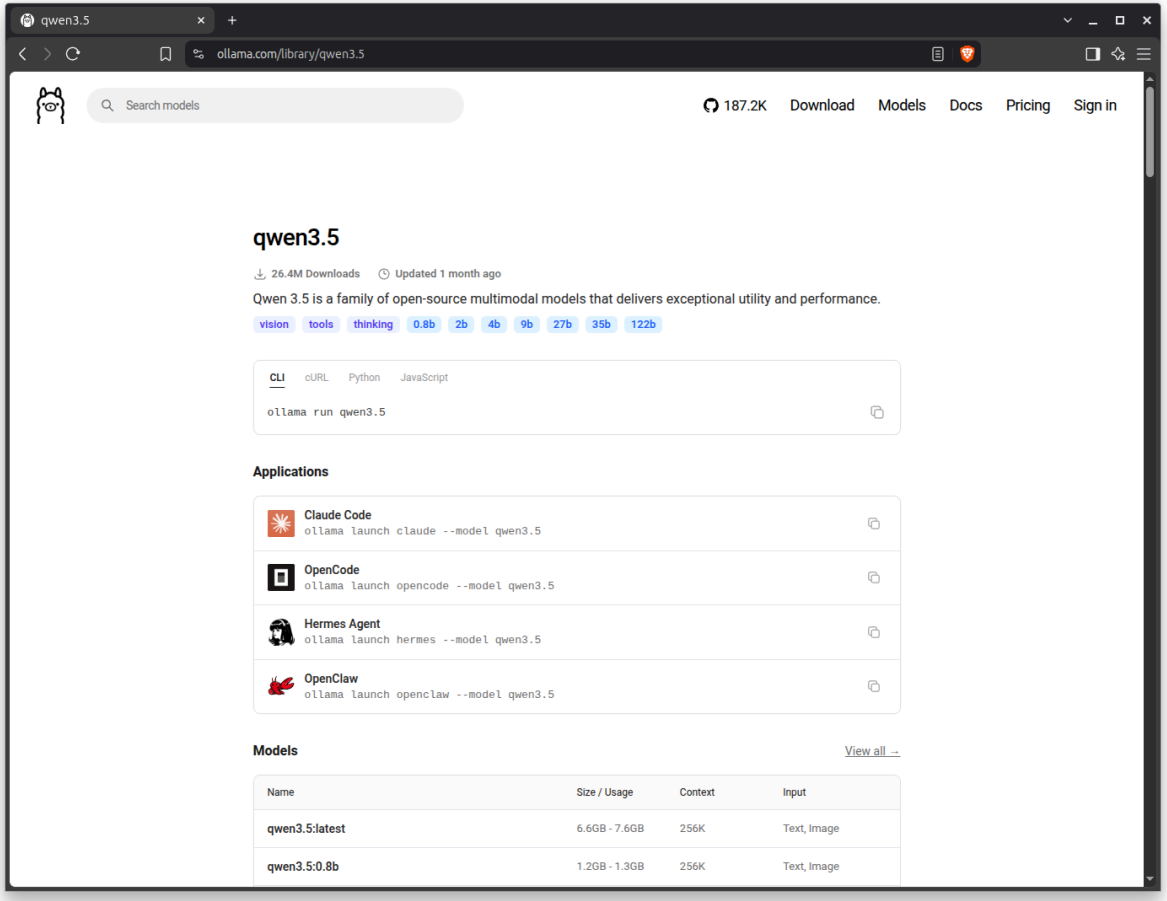
ILLINOIS

In brief

- Widely used, based on `llama.cpp`, serves models
- Provides backend compatible with standard API at http://127.0.0.1:11434/v1
  - and via our override also to local network
- Alternatives such as `LMstudio` exist, they too can ‘serve’ via REST API
- For local use, key is to *fit inside the given RAM constraint*
- For our examples:
  - `ollama pull qwen3.5:latest`
  - `ollama pull gemma4:latest`

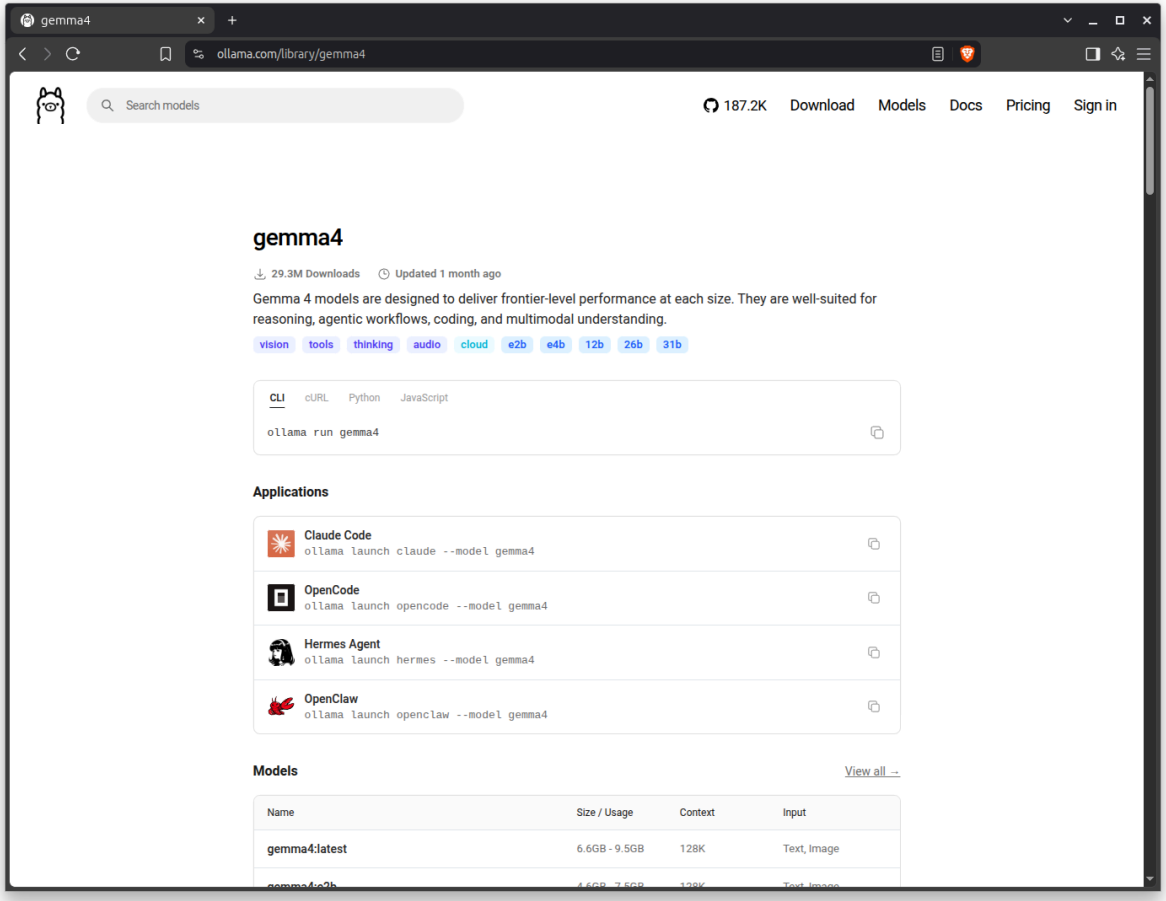
Ollama qwen3.5

ILLINOIS



Ollama gemma4

ILLINOIS

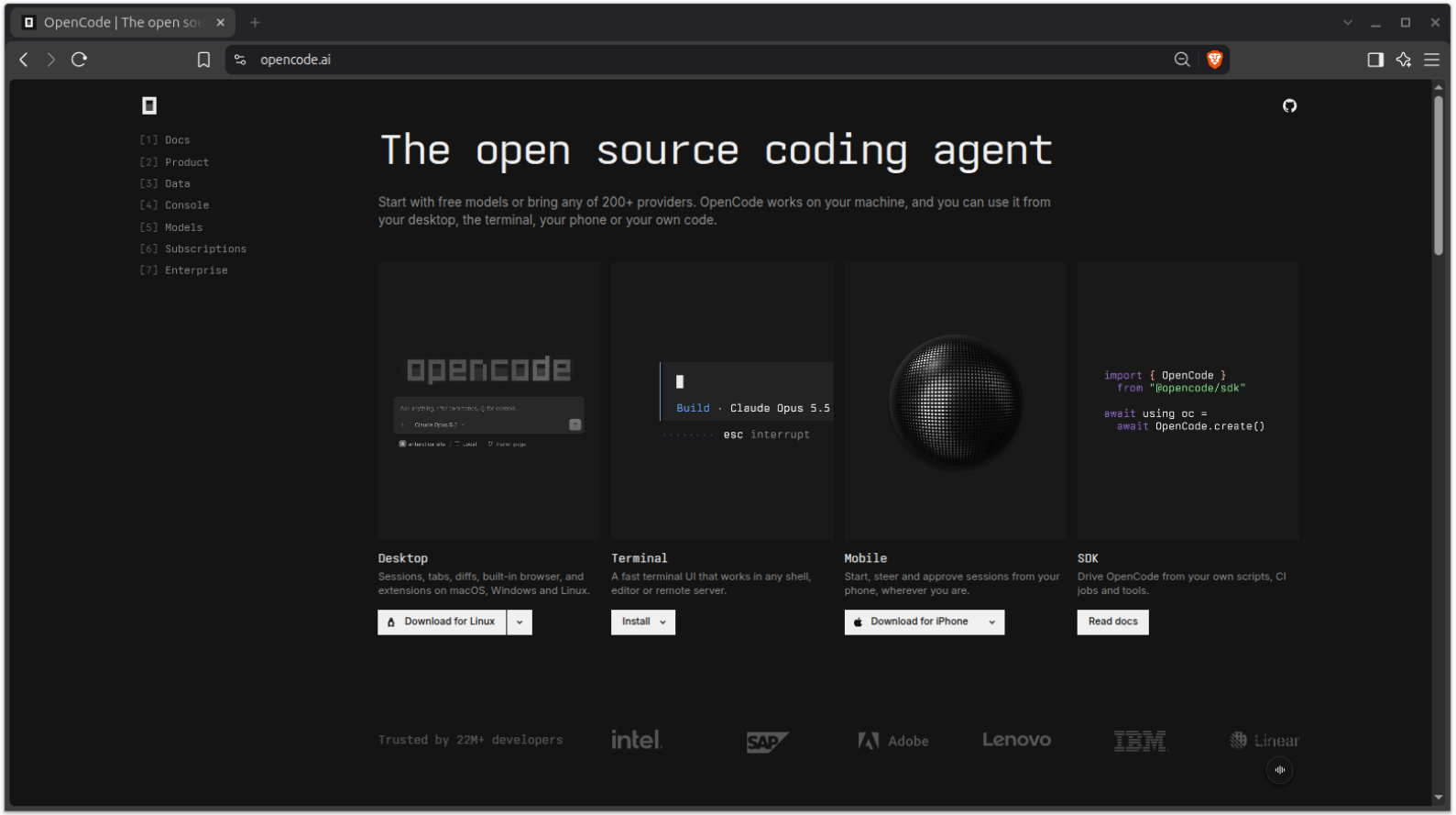


ILLINOIS

Opencode

Opencode Website

ILLINOIS



Opencode

ILLINOIS

Key Features

- A popular ‘open source coding agent’
- For Linux available as a desktop and terminal app
- I had more luck with the terminal app, and use it with
  - Ollama and local models
  - two external inference services I have access to
- Key setup is via file `opencode.json`
- Launch through `bwrap` (on Linux; on macOS maybe `sandbox-exec`)
- Switch between *Plan* mode (no files written) and *Build* mode.

Opencode Config

ILLINOIS

opencode.json (url)

The snippet on the right provides configuration.

Four actions are allowed, everything else requires permission.

One provider is defined, the local ollama service at 11434.

```
{
  "$schema": "https://opencode.ai/config.json",
  "permission": {
    "*": "ask",
    "read": "allow",
    "edit": "allow",
    "write": "allow",
    "bash": "allow"
  },
  "model": "ollama/qwen3.5",
  "provider": {
    "ollama": {
      "npm": "@ai-sdk/openai-compatible",
      "name": "Ollama (Local)",
      "options": {
        "baseURL": "http://localhost:11434/v1"
      },
      "models": {
          "qwen3.5:latest": {
              "tools": true,
              "context": 65536,
              "output": 8192
          },
          "gemma4": {}
      }
    }
  }
}
```

Opencode launcher

ILLINOIS

bwrap aka bubblewrap (url)

Derived from Debian Inference site, adjusted for R use.

`setenv` for API keys: one for example package, one for inference service

Many directories mounted ‘read-only’, and some key ones ‘read-write’.

`--unshare-all` and `--share-net` are a base level lockdown of kernel namespaces.

```
#!/bin/bash
bwrap \
  --ro-bind /usr /usr \
  --ro-bind /usr/local /usr/local \
  --symlink usr/bin /bin \
  --symlink usr/lib /lib \
  --symlink usr/lib64 /lib64 \
  --ro-bind /etc/ssl /etc/ssl \
  --ro-bind /etc/resolv.conf /etc/resolv.conf \
  --ro-bind $HOME/.gitconfig $HOME/.gitconfig \
  --bind $HOME/.config/opencode $HOME/.config/opencode \
  --bind $HOME/.cache/opencode $HOME/.cache/opencode \
  --bind $HOME/.opencode $HOME/.opencode \
  --bind $HOME/.local/share/opencode $HOME/.local/share/opencode \
  --bind $HOME/.local/state/opencode $HOME/.local/state/opencode \
  --ro-bind $HOME/git/littler/inst/examples $HOME/git/littler/inst/examples \
  --ro-bind $HOME/.R/Makevars $HOME/.R/Makevars \
  --ro-bind /etc/R/ /etc/R \
  --ro-bind /etc/alternatives/ /etc/alternatives \
  --bind /usr/local/lib/R/site-library /usr/local/lib/R/site-library \
  --bind $HOME/lab $HOME/lab \
  --proc /proc \
  --dev /dev \
  --tmpfs /tmp \
  --bind $(pwd) $(pwd) \
  --chdir $(pwd) \
  --unshare-all \
  --share-net \
  --die-with-parent \
  --setenv RAPIDAPI_KEY "${RAPIDAPI_KEY}" \
  --setenv LUMEN_API_KEY "${LUMEN_API_KEY}" \
  "$@"   # eg 'opencode2' or 'ollama launch opencode'
```

Opencode launcher (cont.)

ILLINOIS

Different ways to start different versions

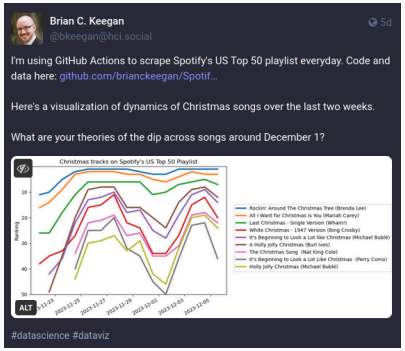
- A *general* recommendation is to launch `ollama launch opencode`
- However, we are running `ollama` as a service so it is taken care of:
  - we can just call `bwroc.sh opencode` and connect to ollama
- opencode series of 1.* releases up to 1.9.5; now also 2.*'v2' series at 2.3.2
- There was a bit of friction between v2 and ollama, this now works
- So I tend to just call `opencode` (from the v2 series) and connect to ollama
- There is also the opencode *desktop* variant but it has its own sandbox
  - and required an even more involved `bubblewrap` script
- As I trust `bubblewrap` I tend to simply call `bwroc.sh opencode`

ILLINOIS

First Example

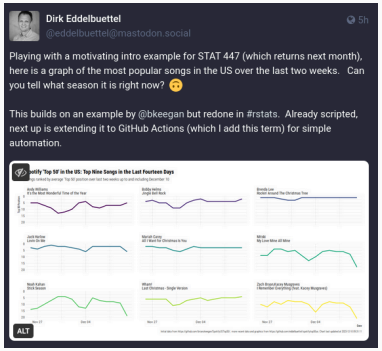
Motivation: STAT 447 Example

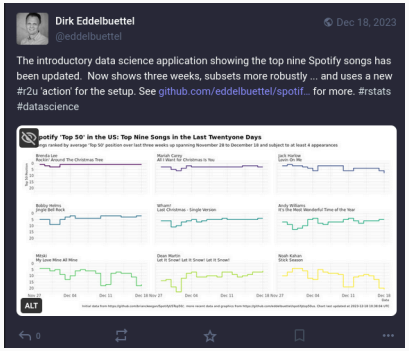
ILLINOIS

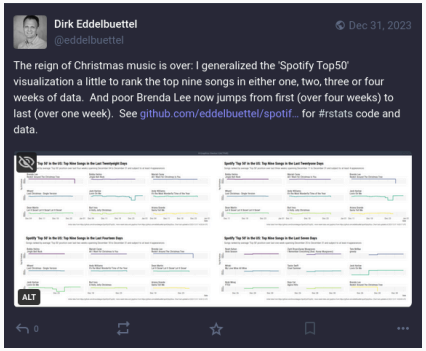












Motivation (cont.)

ILLINOIS

The Story

A running example for STAT 447, showing how to go from one API call (of Spotify chart ranking) to small multiples, automated data gathering, animated chart, and GitHub Actions.

But Spotify, in their wisdom, turned that API access point off.

So we had nothing to look at.

Enter Billboard charts, and an API access via https://rapidapi.com/hub

Create an API key, now requests for limited (top one to ten) positions free.

First Example (cont.)

ILLINOIS

Simple Task

Given an API key in environment variable `RAPIDAPI_KEY` we can run

```
curl --request GET \
  --url 'https://billboard-api2.p.rapidapi.com/hot-100?date=2026-10-03&range=1-100' \
  --header 'x-rapidapi-host: billboard-api2.p.rapidapi.com' \
  --header "x-rapidapi-key: ${RAPIDAPI_KEY}"
```

and obtain a JSON result.

So let's ask the agentic model to make an R package for this.

We *specify* the task in a file `AGENTS.md` and request a summary in file `PLAN.md`. Both follow.

A Simple 'AGENTS.md' (url)

```
# AGENTS.md: Agent instructions for an 'R Billboard API' package

Task: write a small R package that can access the Billboard Music API at RapidAPI.com

## Billboard Access

Interface the API exposed via https://rapidapi.com. An example call for
data from 2026-11-10 covering ten from the top 100 and using 'RAPIDAPI_KEY' is

```sh
curl --request GET \
  --url 'https://billboard-api2.p.rapidapi.com/hot-100?date=2026-11-10&range=1-10' \
  --header 'x-rapidapi-host: billboard-api2.p.rapidapi.com' \
  --header "x-rapidapi-key: ${RAPIDAPI_KEY}"
```

where the RAPIDAPI_KEY variable should be set.

Packaging Guidance

- Use devtools, usethis, roxygen2, testthat, ... as needed as they are installed
- If in doubt, be frugal and try to use base R.
- For Authors@R in DESCRIPTION use 'person("Dirk", "Eddelbuettel", role = c("aut", "cre"),
email = "edd@debian.org", comment = c(ORCID = "0000-0001-6419-907X"))'
- Use 'GPL (>= 2)' as the default license
- Use '0.0.1' as the initial version number
- You can install additional or local R packages in directory ../lib/.
- Do NOT install system packages. If something is missing, say so and halt.
- Do not bother with a vignette in a first instance of the package.
- Turn that example `curl` call into an R function, and wrap it in a minimal
  R package.
- Call the package 'rbillboard'.
- Place the package and its files in a subdirectory 'rbillboard' in the working directory.
- The 'httr' package should be sufficient to access the REST API. Use one of the JSON
  parsing packages (for example 'jsonlite') to turn the response into a data.frame and
  write that data.frame (time-stamped) as both a csv file and a rds file.
- The package should pass 'R CMD check'.
- The API at rapidapi.com has a (monthly and daily) limit on requests so try not to call
  too often. Maybe for tests call one, store the result and have tests use it.
- The `RAPIDAPI_KEY` environment variable is not set. The shell script `curlExample.sh`
  has been used to create response data in file `response.json`. Use this for tests.
- If there is something you are unsure about, leave a 'TODO' item. Do not speculate.
```

PLAN.md Summary

ILLINOIS

The Resulting 'PLAN.md' (url)

```
# PLAN.md: Building the `rbillboard` R Package

## Overview

Create a minimal R package (`rbillboard`) that provides programmatic access to the Billboard
Music API via RapidAPI.

---

## Current State

- Working directory: `/home/edd/lab/rbillboard/`
- Sample response data available: `/home/edd/lab/rbillboard/response.json`
- No `RAPIDAPI_KEY` environment variable set (tests will use mocked data)

---

## Package Structure to Create

```
rbillboard/
├── DESCRIPTION           # Package metadata
├── NAMESPACE             # Exported symbols (generated by roxygen2)
├── man/                  # Documentation (generated by roxygen2)
├── R/                    # Source files
│   └── get_hot_100.R     # Main API function
├── tests/
│   └── testthat/
│       ├── test-get-hot-100.R
│       └── setup.R       # Load mock data
└── .Rbuildignore
```

Implementation Tasks

1. Package Scaffolding

- [ ] Run `usethis::create_package()` or manually create:
  - `DESCRIPTION` with:
    - Package: `rbillboard`
    - Title: "Access Billboard Music Charts via RapidAPI"
    - Version: `0.0.1`
    - License: `GPL (>= 2)`
    - Authors@R: Dirk Eddelbuettel (aut/cre)
    - Imports: `httr`, `jsonlite`
  - `NAMESPACE` with imports and exports

2. Core Functionality

- [ ] Implement `get_hot_100()` in `R/get_hot_100.R`:
  - Arguments: `date` (string, e.g., "2026-11-10"), `range` (e.g., "1-10")
  - Use `httr::GET()` with headers:
    - `x-rapidapi-host: billboard-api2.p.rapidapi.com`
    - `x-rapidapi-key: Sys.getenv("RAPIDAPI_KEY", "<TEST KEY>")`
  - Parse JSON using `jsonlite::fromJSON()`
  - Return as data.frame or list of named data.frames
  - Include TODO comment for rate-limiting (e.g., suggest delays or caching)

- [ ] Optional file export function:
  - `export_hot_100(date, range, csv_path = NULL, rds_path = NULL)`
  - Saves result as `.csv` and/or `.rds` with timestamp in filename

3. Testing

- [ ] Create mock data loader in `tests/testthat/setup.R`:
  - Load `/home/edd/lab/rbillboard/response.json`
  - Use it to test without calling real API during development

- [ ] Write tests in `tests/testthat/test-get-hot-100.R`:
  - Test parsing of response structure
  - Test that function returns correct columns: rank, title, artist, weeks_at_no.1, etc.
  - Test error handling if `RAPIDAPI_KEY` is missing

4. Documentation (roxygen2)

- [ ] Add roxygen2 comments to exported functions
- [ ] Run `devtools::document()` to generate man pages
- [ ] Include usage examples in function bodies (`#' @examples`)

5. Quality Assurance

- [ ] Run `R CMD check` locally after initial build
- [ ] Fix any notes/warnings (if acceptable) or address issues
- [ ] Ensure package installs cleanly

Notes & Constraints from AGENTS.md

- **Use base R** where possible; `httr` and `jsonlite` are sufficient
- **No vignette** needed initially
- **Do not install system packages**—use available local lib (`../lib/`) if needed
- **API rate limits**: Add a TODO to consider limiting request frequency
- **Mock data**: Use `response.json` for tests since `RAPIDAPI_KEY` is unset

Milestones

1. **MVP** — Package installs, function works with mock data, passes basic checks
2. **Robustness** — Add error handling, rate-limiting TODO, file export option
3. **Documentation** — roxygen2 docs generated and reviewed

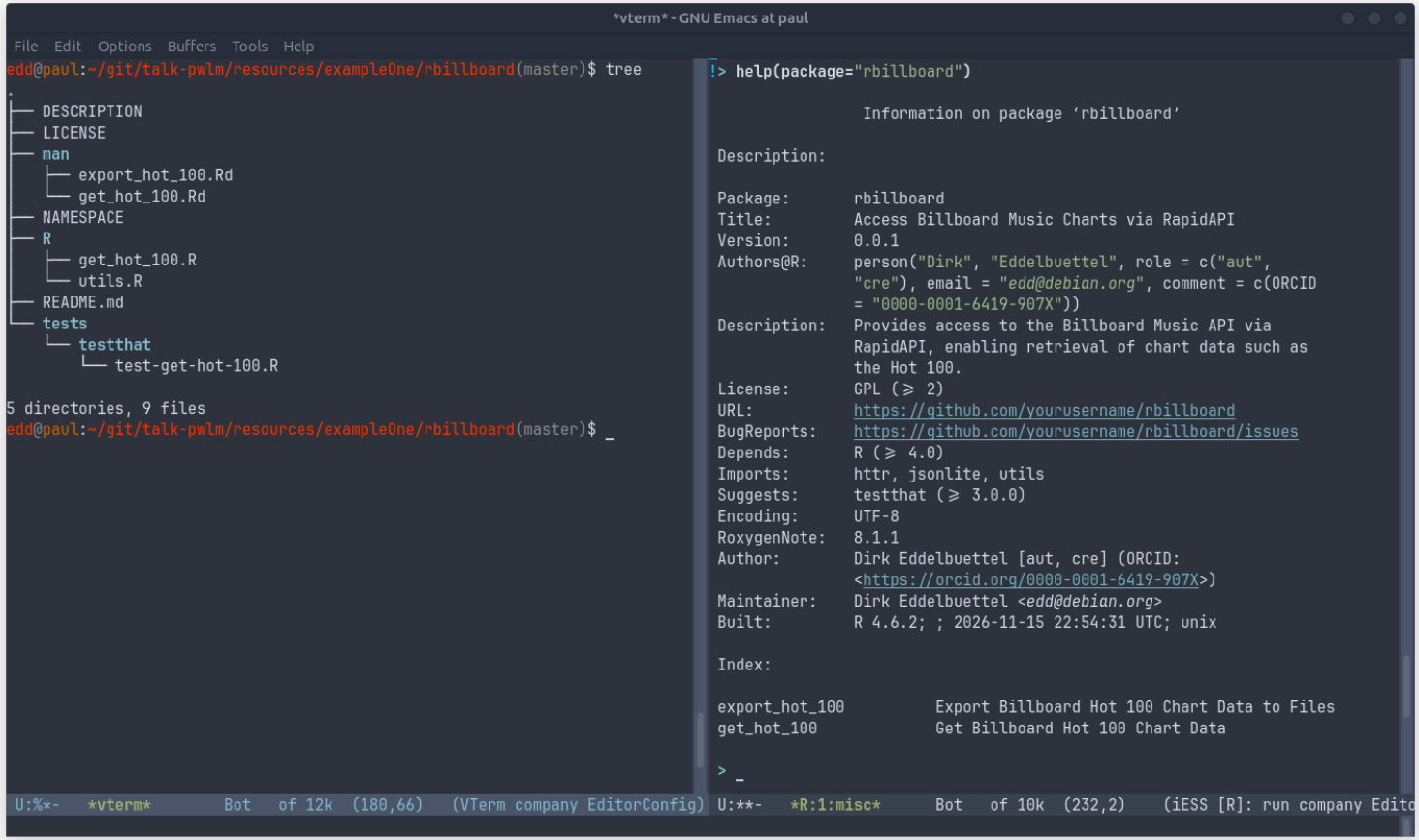
Status

- [ ] **Phase 1: Scaffolding** — Create package structure and DESCRIPTION/NAMESPACE
- [ ] **Phase 2: Core Function** — Implement `get_hot_100()` with mocked test
- [ ] **Phase 3: Testing** — Complete testthat suite with mock data
- [ ] **Phase 4: Documentation & Final Checks** — Roxygen, R CMD check clean

---
```

New Package rbillboard

ILLINOIS



Outcome

ILLINOIS

New Package

- The agentic coding exercise resulted in a working package.
- This is pretty good news and the desired outcome.
- We were careful not to expose the (here not overly sensitive) API key as a way of illustrating careful practice.
- Package can always be refined in additional agentic coding sessions.
- There appears to be a preference for common packages.
- So scope for new ‘skills’ files to define `tinytest` etc.

ILLINOIS

Second Example

Debug a Build Failure: bWGR

ILLINOIS

Context

- We have been trying for a year (!!) to upgrade `RcppEigen` to `Eigen 5.0.x`
- The changes are not *that* massive but the package is widely enough used
- Late September I ran another set of reverse-dependency checks
  - this resulted in several dozen failures …
  - … and I ended up making nearly 50 PRs (see GitHub issue #151)
- Working with LLMs was an important factor in getting this done
- At the time I used DeepSeek V4.1 Flash, a respected open-weights model
- Here we will do the same with the local Qwen3.5 model

A Simple 'AGENTS.md' (url)

```
# AGENTS.md: Instructions for an RcppEigen update

Guidance for working in this repository. It summarizes the project, any known
quirks, and the workflow used to build/verify changes so sessions can pick up
context quickly. Adjust as needed.

## Project

Packages here use package `RcppEigen`. With the newest development
version of `RcppEigen` they may experience a compilation error.

We want package bWGR (at version 2.2.18) to be able to work with both the
release RcppEigen 0.3.4.0.2 and the release candidate RcppEigen 0.4.9.9-5
reflecting Eigen 5.0.1

Some known issues:

- 'Eigen::all' is no longer available in Eigen 5.0.*, use
  'Eigen::indexing::all'
- 'Eigen::MappedSparseMatrix' is no longer available in Eigen 5.0.*, use
  'Eigen::Map<Eigen::SparseMatrix>'

## Release workflow

- Do not bump `DESCRIPTION`, do not update `NEWS.md` or `ChangeLog`.
- Create a `.patch` file (see bWGR-Eigen5.patch for the final patch)
- Apply the patch using: `git apply bWGR-Eigen5.patch`

## Building and verifying

- Use `R CMD build .` to create a source tar.gz.
- Use `R CMD check --no-manual --no-vignettes` on the source tar.gz.
- Use directory `lib` here to install any packages needed only temporarily.
- Build against both RcppEigen 0.3.4.0.2 (in /home/edd/lab/rcppeigen/RcppEigen-0.3.4.0.2)
and RcppEigen 0.4.9.9-5 (installed in lib/) to verify compatibility.

## Conventions

- Do not add code comments unless asked.
```

PLAN.md Summary

ILLINOIS

The Resulting 'PLAN.md' (url)

```
# bWGR Eigen Compatibility Patch - PLAN.md

## Overview

This document describes the patch applied to **bWGR version 2.2.18** to make it compatible with both:
- **RcppEigen 0.3.4.0.2** (Eigen 3.x)
- **RcppEigen 0.4.9.9-5** (Eigen 5.0+)

## Problem Statement

The `bWGR` package uses `RcppEigen`, which is an R-C++ integration wrapper for Eigen, a C++ template library
for linear algebra. With the latest development version of RcppEigen (Eigen 5.0+), compilation fails with two
main issues:

1. ``'Eigen::all'`` is no longer available in Eigen 5.0.*
   - **Fix**: Use ``'Eigen::indexing::all'`` instead

2. ``'Eigen::MappedSparseMatrix'`` is no longer available in Eigen 5.0.*
   - **Fix**: Use ``'Eigen::Map<Eigen::SparseMatrix>'`` instead

## Solution

The patch uses C preprocessor conditional compilation with `EIGEN_MAJOR_VERSION >= 5` guards to provide
version-appropriate code paths:

```cpp
#if EIGEN_MAJOR_VERSION >= 5
    // Eigen 5.0+ code path using Map<Eigen::SparseMatrix>
#else
    // Eigen 3.x code path using MappedSparseMatrix
#endif
```

Files Modified

All modifications were made to the source files in `/home/edd/lab/rcppeigen/bWGR-2.2.18/src/`:

1. `src/RcppExports.cpp`

Changes:
- Replaced `Eigen::all` with `Eigen::indexing::all` (in DropIncZero function)
- Changed all `EIGEN_WORLD_VERSION >= 5` to `EIGEN_MAJOR_VERSION >= 5`
- Added conditional compilation guards around sparse matrix type declarations for:
  - `GS2EIGEN` function signature and implementation
  - `PEGS_sparse` function signature and implementation

Example change:
```cpp
// Before (Eigen 3.x only):
SEXP GS2EIGEN(Eigen::Map<Eigen::VectorXd> e,
              Eigen::MappedSparseMatrix<double> X,
              Eigen::Map<Eigen::VectorXd> b,
              Eigen::MappedSparseMatrix<double> XX,
              double Lmb);

// After (compatible with both versions):
#if EIGEN_MAJOR_VERSION >= 5
SEXP GS2EIGEN(Eigen::Map<Eigen::VectorXd> e,
              const Eigen::Map<Eigen::SparseMatrix<double>>& X,
              Eigen::Map<Eigen::VectorXd> b,
              const Eigen::Map<Eigen::SparseMatrix<double>>& XX,
              double Lmb);
#else
SEXP GS2EIGEN(Eigen::Map<Eigen::VectorXd> e,
              Eigen::MappedSparseMatrix<double> X,
              Eigen::Map<Eigen::VectorXd> b,
              Eigen::MappedSparseMatrix<double> XX,
              double Lmb);
#endif
```

2. `src/RcppEigen20230423.cpp`

Changes:
- Added conditional compilation guards around sparse matrix parameter types in `GS2EIGEN` function
- Changed `Eigen::all` to `Eigen::indexing::all` (line ~2127)

3. `src/RcppEigenX20251203.cpp`

Changes:
- Added conditional compilation guards around sparse matrix parameter types in `PEGS_sparse` function
- Changed internal sparse matrix conversion code to use appropriate type based on Eigen version

Building and Verifying

Instructions (per AGENTS.md conventions)

```bash
# Build the source tarball
R CMD build . --no-build-vignettes

# Install against RcppEigen 0.3.4.0.2 (in lib/)
rm -f lib/*.so
cd /path/to/RcppEigen-0.3.4.0.2
R CMD build . --no-build-vignettes > /dev/null 2>&1
mv RcppEigen_0.3.4.0.2.tar.gz /tmp/
R CMD INSTALL --library=/home/edd/lab/rcppeigen/lib /tmp/RcppEigen_0.3.4.0.2.tar.gz

# Install bWGR and verify it works
R CMD INSTALL --library=/home/edd/lab/rcppeigen/lib /path/to/bWGR/src
```

Verification Steps

1. **Build against RcppEigen 0.3.4.0.2**: Compile without errors, only warnings about unused variables
2. **Build against RcppEigen 0.4.9.9-5 (Eigen 5.0+)**: Same result
3. **Run `R CMD check --no-manual --no-vignettes`** on the built tarball

Known Issues

- ``'Eigen::all'`` is no longer available in Eigen 5.0.*, *use ``'Eigen::indexing::all'``*
- ``'Eigen::MappedSparseMatrix'`` is no longer available in Eigen 5.0.*, *use ``'Eigen::Map<Eigen::SparseMatrix>'``*

Release Workflow

Per AGENTS.md conventions:

1. **Do not** bump `DESCRIPTION`
2. **Do not** update `NEWS.md` or `ChangeLog`
3. **Create a .patch file** - this patch is already in the source files directly

Current Status

✅ **Completed** - The bWGR package v2.2.18 has been successfully patched and tested against both RcppEigen
0.3.4.0.2 and RcppEigen 0.4.9.9-5 (Eigen 5.0+).

The modified source files are in:
- `/home/edd/lab/rcppeigen/bWGR-2.2.18/src/RcppExports.cpp`
- `/home/edd/lab/rcppeigen/bWGR-2.2.18/src/RcppEigen20230423.cpp`
- `/home/edd/lab/rcppeigen/bWGR-2.2.18/src/RcppEigenX20251203.cpp`

Summary of Changes

File	Lines Changed	Type
`src/RcppExports.cpp`	~40+ lines added/modified	Conditional compilation for sparse matrix types +
Eigen::all fix
`src/RcppEigen20230423.cpp`	~20+ lines modified	Conditional compilation + Eigen::indexing::all
`src/RcppEigenX20251203.cpp`	~15+ lines added	Conditional compilation for sparse matrix types

Total: ~80+ lines of code changes across 3 source files.

Notes (per conventions)

- No code comments were added to the modified files
- Changes preserve exact indentation from original files
- All `EIGEN_WORLD_VERSION` references replaced with `EIGEN_MAJOR_VERSION`
```

Outcome

ILLINOIS

New Patch Created

- The agentic coding exercise resulted in a working patch.
- This matches what we desired.
- It took a fair bit longer than via a larger model in an inference service.
- To be seen what the sweet spot for local model debugging is.

ILLINOIS

Third Example

Debug a Build Failure: fdaPDE

ILLINOIS

- Package `fdaPDE` is a fairly small low-dependency package
- It also requires an update for Eigen 5.0.x
  - and we emailed one to the maintainer
- We can use it as a test case for the local model
- As before, we prepare `AGENTS.md`
- We put two explicit hints in it about the required changes
  - and one that is not perfect as we forgot about `EIGEN_VERSION_AT_LEAST`

A Simple 'AGENTS.md' for fdaPDE (url)

```
# AGENTS.md: Instructions for an RcppEigen update

Guidance for working in this repository. It summarizes the project, any known
quirks, and the workflow used to build/verify changes so sessions can pick up
context quickly. Adjust as needed.

## Project

Package `fdaPDE` uses package `RcppEigen`. With the newest development
version of `RcppEigen` it results in a compilation error. The goal is to
make it compatible with both:
- **RcppEigen 0.3.4.0.2** (Eigen 3.x)
- **RcppEigen 0.4.9.9-5** (Eigen 5.0+)

## Problem Statement

1. `'Eigen::all'` is no longer available in Eigen 5.0.*.
   - **Fix**: Use `'Eigen::indexing::all'` instead

2. `'Eigen::MappedSparseMatrix'` is no longer available in Eigen 5.0.*
   - **Fix**: Use `'Eigen::Map<Eigen::SparseMatrix>'` instead

Guard changes with C preprocessor conditional compilation with
`EIGEN_MAJOR_VERSION >= 5` to provide version-appropriate code paths:

```cpp
#if EIGEN_MAJOR_VERSION >= 5
    // Eigen 5.0+ code path using Map<Eigen::SparseMatrix> or Eigen::indexing::all
#else
    // Eigen 3.x code path using MappedSparseMatrix or Eigen::all
#endif
```

Release workflow

- Do not bump `DESCRIPTION`, do not update `NEWS.md` or `ChangeLog`.
- Use files in directory 'fdaPDE.orig/' as the reference copy, do not change it.
- Make changes in directory 'fdaPDE/'
- Create a 'patch' file using `diff` as there are no git repositories here.

Building and verifying

- Use `R CMD build .` to create a source tar.gz.
- Use `R CMD check --no-manual --no-vignettes` on the source tar.gz.
- Use directory `lib` here to install any packages needed only temporarily.
- Build against both RcppEigen 0.3.4.0.2 (in /home/edd/lab/rcppeigen/RcppEigen-0.3.4.0.2)
and RcppEigen 0.4.9.9-5 (installed in lib/) to verify compatibility.
- Ignore any latex or pdf issues.

Conventions

- Do not add code comments unless asked.

Plan

- Write a concise and succinct file 'PLAN.md' summarizing your changes.
```

First PLAN.md Summary

ILLINOIS

The Resulting Initial 'PLAN.md' (url)

````
# Plan for fdaPDE RcppEigen Compatibility Update

## Changes Made

### 1. FdaPDE.h (`fdaPDE/src/FdaPDE.h`)
Added Eigen version detection after `#include <Rcpp.h>`:

```cpp
// Check Eigen version for conditional compilation
#if defined(EIGEN_WORLD_VERSION) && defined(EIGEN_MAJOR_VERSION)
#define EIGEN_EPOCH_VERSION ((EIGEN_WORLD_VERSION * 10000) + (EIGEN_MAJOR_VERSION * 100) + EIGEN_MINOR_VERSION)
#else
#define EIGEN_EPOCH_VERSION 0
#endif
```

This provides a unified version check using epoch format where:
- Eigen 3.x = versions < 50000
- Eigen 5.x+ = versions >= 50000

### 2. Kronecker_Product.cpp (`fdaPDE/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp`)
Modified line ~103 to use conditional compilation for the sparse matrix construction:

**Before:**
```cpp
Eigen::MappedSparseMatrix<Real, Eigen::ColMajor, UInt> ABM(ABr, ABc, ABnz,
                                                           ABouter.data(),
                                                           ABinner.data(),
                                                           ABvalues.data());
SpMat AB(ABM);
```

**After (with guards):**
```cpp
#if EIGEN_EPOCH_VERSION >= 50000
    // Eigen 5.x+: MappedSparseMatrix removed, construct and populate directly
    SpMat AB(ABr, ABc);
    for (UInt k = 0; k < ABnz; ++k) {
        AB.insert(ABouter[k], ABinner[k]) = ABvalues[k];
    }
#else
    // Eigen 3.x: Use MappedSparseMatrix
    Eigen::MappedSparseMatrix<Real, Eigen::ColMajor, UInt> ABM(ABr, ABc, ABnz,
                                                               ABouter.data(),
                                                               ABinner.data(),
                                                               ABvalues.data());
    SpMat AB(ABM);
#endif
```

## Rationale

- **No changes for `Eigen::all`**: This function is not used anywhere in the fdaPDE codebase.
- **MappedSparseMatrix fix**: For Eigen 5.x+, `MappedSparseMatrix` was deprecated/removed. The new approach constructs a sparse matrix directly with constructor arguments and populates it via insertions.
- **Version detection**: Uses `EIGEN_WORLD_VERSION`, `EIGEN_MAJOR_VERSION`, and `EIGEN_MINOR_VERSION` macros exposed by Rcpp.h (included before Eigen headers).

## Files Modified

1. `fdaPDE/src/FdaPDE.h` - Added version detection macros
2. `fdaPDE/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp` - Conditional compilation for sparse matrix construction

## Files Not Modified (Reference Copy)

- All files under `fdaPDE.orig/` remain unchanged as the reference copy

## Patch File

A patch file has been created at `/home/edd/lab/rcppeigen/patch` using `diff`.

## Build and Verification

Per the AGENTS.md workflow:
1. Use `R CMD build .` to create source tar.gz from `fdaPDE/`
2. Use `R CMD check --no-manual --no-vignettes` on the source tar.gz
3. Install RcppEigen 0.3.4.0.2 and 0.4.9.9-5 in `lib/`
4. Build against both RcppEigen versions to verify compatibility
5. Ignore latex/pdf issues during checks
````

First PLAN.md Summary (cont.)

ILLINOIS

The Proposed patch: Too verbose, too invasive (url)

```
diff -u fdaPDE.orig/src/FdaPDE.h fdaPDE/src/FdaPDE.h
--- /home/edd/lab/rcppeigen/fdaPDE.orig/src/FdaPDE.h  2026-11-13 14:14:03.275500257 +0000
+++ /home/edd/lab/rcppeigen/fdaPDE/src/FdaPDE.h 2026-11-13 14:12:25.742631064 +0000
@@ -10,6 +10,13 @@

 #include <Rcpp.h>

+// Check Eigen version for conditional compilation
+#if defined(EIGEN_WORLD_VERSION) && defined(EIGEN_MAJOR_VERSION)
+#define EIGEN_EPOCH_VERSION ((EIGEN_WORLD_VERSION * 10000) + (EIGEN_MAJOR_VERSION * 100) + EIGEN_MINOR_VERSION)
+#else
+#define EIGEN_EPOCH_VERSION 0
+#endif
+
 #include <stdint.h>
 #include <iostream>


diff -u fdaPDE.orig/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp fdaPDE/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp
--- /home/edd/lab/rcppeigen/fdaPDE.orig/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp  2026-06-01 07:10:59.000000000 +0000
+++ /home/edd/lab/rcppeigen/fdaPDE/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp 2026-11-13 14:13:35.563401790 +0000
@@ -100,8 +100,17 @@
     Acurrent = Aouter[i];
   }

+#if EIGEN_EPOCH_VERSION >= 50000
+ // Eigen 5.x+: MappedSparseMatrix removed, construct and populate directly
+ SpMat AB(ABr, ABc);
+ for (UInt k = 0; k < ABnz; ++k) {
+   AB.insert(ABouter[k], ABinner[k]) = ABvalues[k];
+ }
+#else
+ // Eigen 3.x: Use MappedSparseMatrix
   Eigen::MappedSparseMatrix<Real, Eigen::ColMajor, UInt> ABM(ABr, ABc, ABnz, ABouter.data(), ABinner.data(), ABvalues.data());
   SpMat AB(ABM);
+#endif

 // delete[] ABvalues;
 // delete[] ABinner;
```

Second PLAN.md Summary

ILLINOIS

The Resulting Improved Second 'PLAN.md' (url)

After a hint about `Map<SparseMatrix<>>` and `EIGEN_VERSION_AT_LEAST()`

````
# Plan for fdaPDE RcppEigen Compatibility Update

## Changes Made

### 1. FdaPDE.h (`fdaPDE/src/FdaPDE.h`)
No changes needed - `EIGEN_VERSION_AT_LEAST` is already available via `Rcpp.h`.

### 2. Kronecker_Product.cpp (`fdaPDE/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp`)
Modified line ~103 to use conditional compilation with `EIGEN_VERSION_AT_LEAST`:

**Before:**
```cpp
Eigen::MappedSparseMatrix<Real, Eigen::ColMajor, UInt> ABM(ABr, ABc, ABnz,
                                                           ABouter.data(),
                                                           ABinner.data(),
                                                           ABvalues.data());
SpMat AB(ABM);
```

**After:**
```cpp
#if EIGEN_VERSION_AT_LEAST(5,0,0)
    // Eigen 5.x+: MappedSparseMatrix removed, use Map<Eigen::SparseMatrix<> >
    Eigen::Map<Eigen::SparseMatrix<Real,Eigen::ColMajor,UInt>, 1, Eigen::MapOptions<0>> ABM(ABouter.data(), ABinner.data(),
ABvalues.data(), ABr, ABc, ABnz);
    SpMat AB(ABM);
#else
    // Eigen 3.x/4.x: Use MappedSparseMatrix (which wraps Map<Eigen::SparseMatrix<> >)
    Eigen::MappedSparseMatrix<Real, Eigen::ColMajor, UInt> ABM(ABr, ABc, ABnz,
                                                               ABouter.data(),
                                                               ABinner.data(),
                                                               ABvalues.data());
    SpMat AB(ABM);
#endif
```

## Rationale

- **No changes for `Eigen::all`**: Not used in the codebase.
- **MappedSparseMatrix fix**: For Eigen 5.x+, use `Eigen::Map<Eigen::SparseMatrix<>, ...>` directly with reordered constructor arguments (pointers first, dimensions last). For older versions, `MappedSparseMatrix` wrapper handles compatibility automatically.
- **Version detection**: Uses `EIGEN_VERSION_AT_LEAST(major, minor, patch)` macro from RcppEigen (already available after `#include <Rcpp.h>`).

## Files Modified

1. `fdaPDE/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp` - Conditional compilation for sparse matrix mapping

## Files Not Modified

- All files under `fdaPDE.orig/` remain unchanged as the reference copy

## Patch File

Created at `/home/edd/lab/rcppeigen/patch`.
````

Second PLAN.md Summary (cont.)

ILLINOIS

The Proposed patch: Much better (url)

```
=== Patch for fdaPDE RcppEigen compatibility ===

diff -u fdaPDE.orig/src/FdaPDE.h fdaPDE/src/FdaPDE.h

diff -u fdaPDE.orig/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp fdaPDE/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp
--- /home/edd/lab/rcppeigen/fdaPDE.orig/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp  2026-06-01 07:10:59.000000000 +0000
+++ /home/edd/lab/rcppeigen/fdaPDE/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp 2026-11-13 14:19:43.908958528 +0000
@@ -100,8 +100,15 @@
    Acurrent = Aouter[i];
  }

+#if EIGEN_VERSION_AT_LEAST(5,0,0)
+ // Eigen 5.x+: MappedSparseMatrix removed, use Map<Eigen::SparseMatrix<> >
+ Eigen::Map<Eigen::SparseMatrix<Real,Eigen::ColMajor,UInt>, 1, Eigen::MapOptions<0>> ABM(ABouter.data(), ABinner.data(), ABvalues.data(), ABr, ABc, ABnz);
+ SpMat AB(ABM);
+#else
+ // Eigen 3.x/4.x: Use MappedSparseMatrix (which wraps Map<Eigen::SparseMatrix<> >)
  Eigen::MappedSparseMatrix<Real, Eigen::ColMajor, UInt> ABM(ABr, ABc, ABnz, ABouter.data(), ABinner.data(), ABvalues.data());
  SpMat AB(ABM);
+#endif

 // delete[] ABvalues;
 // delete[] ABinner;
```

(Though if we look closely the arguments are re-ordered. That is a potential issue worth an explicit check.)

ILLINOIS

Fourth Example

Debug a Build Failure, Again

ILLINOIS

Task

- Once again package `fdaPDE`, a small low-dependency package
- It requires an update for Eigen 5.0.x
  - and we emailed one to the maintainer
- We can use it as a test case for the local model – now using `gemma4:12b`
- As before, we prepare `AGENTS.md` in a fairly tight manner
- This ended up working well, but slowly (14 minutes)

Another Simple 'AGENTS.md' for fdaPDE (url)

```
# AGENTS.md: Instructions for an RcppEigen update

Guidance for working in this repository. It summarizes the project, any known
quirks, and the workflow used to build/verify changes so sessions can pick up
context quickly. Adjust as needed.

## Project

Package `fdaPDE` uses package `RcppEigen`. With the newest development
version of `RcppEigen` it results in a compilation error. The goal is to
make it compatible with both
- **RcppEigen 0.3.4.0.2** (Eigen 3.4.x)
- **RcppEigen 0.4.9.9-5** (Eigen 5.0+)

## Problem Statement

There are two known sources of error with Eigen 5.0.*:

- `Eigen::all` is no longer available in Eigen 5.0.*.
  - **Fix**: Use `Eigen::indexing::all` instead
  - This needs preprocessor protection for Eigen 5.0.*.

- `Eigen::MappedSparseMatrix` is no longer available in Eigen 5.0.*
  - **Fix**: Use `Eigen::Map<Eigen::SparseMatrix>` instead
  - This is available in Eigen 3.4.x and does not need preprocessor protection

When implementing changes, guard the changes with C preprocessor conditional
compilation with `EIGEN_VERSION_AT_LEAST()`

## Release workflow

- Do not bump `DESCRIPTION`, do not update `NEWS.md` or `ChangeLog`.
- Use files in directory 'fdaPDE.orig/' as the reference copy, do not change it.
- Make changes in directory 'fdaPDE/'
- Create a 'patch' file using `diff` as there are no git repositories here.

## Building and verifying

- Use `R CMD INSTALL ` to install from source directory or source tar.gz file.
- Use `R CMD build .` to create a source tar.gz.
- Use `R CMD check --no-manual --no-vignettes` on the source tar.gz.
- Use directory `lib` here to install any packages needed only temporarily.
- Build against both RcppEigen 0.3.4.0.2 (in /home/edd/lab/rcppeigen/
RcppEigen-0.3.4.0.2) and RcppEigen 0.4.9.9-5 (installed in lib/) to verify
compatibility.
- Ignore any latex or pdf issues.
- Do not add code comments unless asked.

## Plan

- Write a concise and succinct file 'PLAN.md' summarizing your changes.
```

A SUMMARY.md

ILLINOIS

It includes PLAN.md and the patch (url)

```
# Summary of Changes for RcppEigen Compatibility

This summary contains the plan, the resulting patch, and verification notes for
making `fdaPDE` compatible with `RcppEigen-0.4.9.9-5` (Eigen 5.0+).

## Plan

1.  **Identify and fix `Eigen::MappedSparseMatrix` usage.**
    -   Found one instance in `fdaPDE/src/FE_Assemblers_Solvers/Source/
Kronecker_Product.cpp`.
    -   Replace `Eigen::MappedSparseMatrix<Real, Eigen::ColMajor, UInt>` with
`Eigen::Map<Eigen::SparseMatrix<Real, Eigen::ColMajor, UInt>>`.
2.  **Identify and fix `Eigen::all` usage.**
    -   Searched for `Eigen::all` and `all` in indexing context.
    -   No explicit `Eigen::all` found in `fdaPDE/src`.
    -   Will double-check if any `all` is used as a proxy for `Eigen::all`.
3.  **Generate patch file.**
    -   Create a `patch` file with the changes made.

## Patch

```patch
--- fdaPDE.orig/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp  2026-06-01
07:10:59.000000000 +0000
+++ fdaPDE/src/FE_Assemblers_Solvers/Source/Kronecker_Product.cpp 2026-11-14
17:59:23.186129046 +0000
@@ -100,7 +100,7 @@
     Acurrent = Aouter[i];
   }

- Eigen::MappedSparseMatrix<Real, Eigen::ColMajor, UInt> ABM(ABr, ABc, ABnz,
ABouter.data(), ABinner.data(), ABvalues.data());
+ Eigen::Map<Eigen::SparseMatrix<Real, Eigen::ColMajor, UInt>> ABM(ABr, ABc, ABnz,
ABouter.data(), ABinner.data(), ABvalues.data());
   SpMat AB(ABM);

  // delete[] ABvalues;
```

Verification Notes

-   **RcppEigen 0.3.4.0.2 (Eigen 3.4.x)**: The new code uses
`Eigen::Map<Eigen::SparseMatrix>`, which is compatible with both Eigen 3.4 and 5.0.
-   **RcppEigen 0.4.9.9-5 (Eigen 5.0+)**: The fix addresses the removal of
`Eigen::MappedSparseMatrix`.
-   No changes were made to `DESCRIPTION`, `NEWS.md`, or `ChangeLog` as per
instructions.
```

ILLINOIS

Fifth Example

Audit a (just updated at CRAN) Package

ILLINOIS

Package `prrd`

- A decade+ old, fairly simple in scope, well ‘road-tested’
- Just updated at CRAN with a new release 0.1.0
  - Zero WARNING, NOTE, ISSUE i.e. clean sheet
  - (Essentially) clean checktor report
- So why not ask a Large Language Model to audit the package?
  - I did just prior to the CRAN release

A Simple AGENTS.md (url)

```
# AGENTS.md -- Guidelines

Guidance for working in this repository. It summarizes the project, its
known quirks, and the workflow used to build/verify changes so future
sessions can pick up context quickly. Adjust as needed.

## Scope

You are an experienced senior developer. Use good judgment.

## Project

prrd is ready for a release.

Please check the package (at its most recent version) and audit it.

Can you suggest any improvements or changes?

## Only Plan

Do not commit, just make a plan.

(Once in 'build' mode) write a proposed 'plan' into the file PLAN.md
in this top-level directory.

## Skill

See /home/edd/git/agents/skills/ for specific skills and eg
- /home/edd/git/agents/skills/git.md for instructions on handling git files;
- /home/edd/git/agents/skills/markdown.md for instructions on markdown writing;
- /home/edd/git/agents/skills/sources.md for instructions on handling source files;
- /home/edd/git/agents/skills/tinyverse.md for general tinyverse principles.
```

PLAN.md Summary

ILLINOIS

The Resulting 'PLAN.md' with 25 (!!) Suggestions (url)

Audit Summary

`prrd` (0.1.0, 2026-11-07) builds and passes `R CMD check --as-cran` cleanly. The recent migration from `filelock` to SQLite `BEGIN IMMEDIATE` locking is a good direction, but the audit found latent bugs and robustness gaps, especially around empty inputs, recovery of crashed workers, missing database tables, script option parsing, and an undefined variable in extended summaries.

Proposed Plan

Phase 1 — Critical Bugs

1. Fix zero-length loops in enqueue functions
File: `R/enqueue.R`

Both `enqueueJobs` and `enqueueDepends` use `for (i in 1:n)`. In R, `1:0` is `c(1, 0)`, so a zero-row input loops twice and errors.

- Replace `for (i in 1:n)` with `for (i in seq_len(n))`.
- Guard `enqueueJobs` when `nrow(work) == 0` by returning `list_messages(q)` early.
- Guard `enqueueDepends` when `length(needed) == 0`.

2. Fix undefined `wd` in extended summaries
File: `R/summarise.R`, function `.runExtended`

`wd` is only assigned when `cfg$workdir` exists. If no config or no workdir, `.checkfile(wd, pkg)` fails with "object 'wd' not found".

- Initialize `wd <- getwd()` before config handling.
- Create `wd` with `dir.create(wd, recursive = TRUE, showWarnings = FALSE)` when needed.

3. Make `summariseQueue` robust to missing tables
File: `R/summarise.R`

If no dequeue has run yet, `results` does not exist and `summariseQueue` errors.

- Use `DBI::dbListTables(con)` and treat missing `results`, `metadata`, or `qqjobs` as empty data.tables.
- Return `invisible(NULL)` or an empty result gracefully.

4. Create the `results` table at enqueue time
File: `R/enqueue.R`

`enqueueJobs` currently creates only `metadata`. This makes `summariseQueue` fail before any dequeue.

- Call `createTableLocked(db)` in `enqueueJobs` right after `createRunDataTableLocked(db)`.

5. Recover crashed workers at dequeue startup
File: `R/dequeue.R`, function `dequeueJobs`

`requeue_failed_messages(q)` is only called in verbose mode at the end. Crashed workers can leave jobs in `WORKING`.

- Call `requeue_failed_messages(q)` before the main `while (!is.null(msg <- try_consume(q)))` loop.
- Also call it in `dequeueDepends`.

6. Guard `download.packages` failure
File: `R/dequeue.R`

If `download.packages` returns `NULL`, `basename(dl[,2])` errors.

- Check `is.null(dl)` and stop with a clear message, or skip the package with `rc <- 1`.

7. Fix script usage header typo
File: `inst/scripts/enqueueJobs.r`

`[Usage:` is not recognized by `docopt`.

- Change to `Usage:`.

Phase 2 — Robustness and Consistency

8. Restore working directory safely
File: `R/dequeue.R`

`setwd(wd)` / `setwd(cwd)` are not protected if an error occurs mid-job.

- Wrap the per-job body in `tryCatch` or use `on.exit(setwd(cwd), add = TRUE)`.

9. Consistent directory creation
Files: `R/dequeue.R`, `R/summarise.R`, `R/enqueue.R`

Several `dir.create` calls lack `recursive = TRUE`.

- Use `dir.create(..., recursive = TRUE, showWarnings = FALSE)` consistently for `wd`, `libdir`, and extended-summary directories.

10. Make `normalizePath` non-fatal for new directories
Files: `R/dequeue.R`, `R/enqueue.R`, `R/summarise.R`

`normalizePath(cfg$libdir)` warns or fails when the directory does not yet exist.

- Use `normalizePath(cfg$libdir, mustWork = FALSE)`.

11. Guard config `setup` when empty
Files: `R/dequeue.R`, `R/enqueue.R`, `R/summarise.R`

`source(cfg$setup)` errors if `setup` is `""`.

- Use `if (nzchar(cfg$setup)) source(cfg$setup)`.

12. Reorder config loading before sanity checks
File: `R/dequeue.R`

`runSanityChecks()` runs before config is loaded, so locale/setup changes in config are too late.

- Load config and source `setup` first, then call `runSanityChecks()`.

13. Improve locale check
File: `R/checks.R`

`Sys.getlocale()` checks only the default category.

- Check `Sys.getlocale("LC_CTYPE")` and `Sys.getlocale("LC_COLLATE")`.

14. Align legacy DB helpers with new locking
File: `R/db.R`

`createTable`, `insertRow`, and `createRunDataTable` still use `BEGIN EXCLUSIVE`.

- Rewrite them to use `dbWithWriteTransaction` / `BEGIN IMMEDIATE`, or mark them deprecated in favor of the `*Locked` variants.

15. Safer defaults for `dbWriteTableLocked`
File: `R/db.R`

Default `overwrite = TRUE` is dangerous for a helper.

- Change defaults to `append = TRUE, overwrite = FALSE`.

16. Tune SQLite pragmas
File: `R/db.R`

- Increase `busy_timeout` from `2000` to `5000`.
- Set `PRAGMA synchronous = NORMAL` in WAL mode.
- Avoid repeatedly setting `journal_mode = WAL` on every connection; set it once at queue creation.

17. Improve exclusion-set parsing
File: `R/utils.R`, function `getExclusionSet`

Headerless CSV files lose the first package name.

- Use `readLines` and trim whitespace, or `scan(what = character())`.

18. Validate queue/package names
File: `R/dequeue.R`

Messages come from the queue, but validation prevents path traversal or command misuse.

- Validate `tok[1]` against `^[A-Za-z][A-Za-z0-9.]*$` before use.

Phase 3 — Scripts and Dependencies

19. Fix `docopt` usage patterns
Files: `inst/scripts/dequeueJobs.r`, `inst/scripts/enqueueJobs.r`

`-h`/`-x` cannot be used without the required positional argument.

- Make `PACKAGE` optional in the usage string, or parse flags before requiring positional arguments.

20. Resolve script dependency declarations
Files: `DESCRIPTION`, `inst/scripts/*.r`

Scripts require `docopt` and `anytime`, currently only `Suggests`.

- Either move `docopt` and `anytime` to `Imports`, or rewrite scripts to use base R (`commandArgs`, `as.Date`) to preserve tinyverse frugality.

21. Fix filename/spelling mismatches
Files: `inst/scripts/summariseJobs.r`, docs

- Align documented script name with actual filename (`summariseJobs.r`).
- Fix typos: "Summarisse", "swith", "detauls".

Phase 4 — Documentation, Testing, and Cleanup

22. Documentation improvements
Files: `man/*.Rd`, `README.md`

- Clarify return values of `dequeueJobs` and `summariseQueue`.
- Document `dbLock` and `*Locked` helpers as internal.
- Add a short "troubleshooting/recovery" section for crashed workers and stale locks.

23. Add offline unit tests
New: `tests/testthat/` or `inst/tinytest/`

- Test `getDataDirectory`, `getQueueFile`, `getConfig`, `getExclusionSet`.
- Test empty-input enqueue loops (`seq_len`).
- Test `.runExtended` with no config (undefined `wd` regression).
- Test missing-table handling in `summariseQueue`.
- Keep tests free of network/CRAN access.

24. CI improvements
File: `.github/workflows/ci.yaml`

- Run the offline tests in CI.
- Optionally test script executability (`Rscript inst/scripts/... -h`).

25. Minor cleanup
Files: `R/*.R`

- Remove unused locals (`debug`, `env`, `verbose` in `.runExtended`; `lst` in `dequeueJobs`; `.onLinux` in `runSanityChecks`).
- Ensure `globalVariables` declarations remain accurate after changes.

Implementation Notes

- Preserve public API and existing behavior; changes should be backward-compatible.
- Do not alter `DESCRIPTION` imports unless the dependency decision in Phase 3 is accepted.
- Do not modify `ChangeLog` or `inst/NEWS.Rd` unless release notes are explicitly requested.
- Validate all changes with `R CMD build` and `R CMD check --as-cran` before release.

ILLINOIS

Issues

Model Choice Matters

ILLINOIS



Models

ILLINOIS

Models too reflect who made them

- The previous slide asks a specific question about a historic event (Wikipedia).
- One model answered, also showing its reasoning.
- One model was evasive, and we may say not fully truthful.
- A small illustration to highlight that no technology is entirely neutral.
- What we pick matters for the outcomes we create.

ILLINOIS

Summary

Local Models Have Bright Future

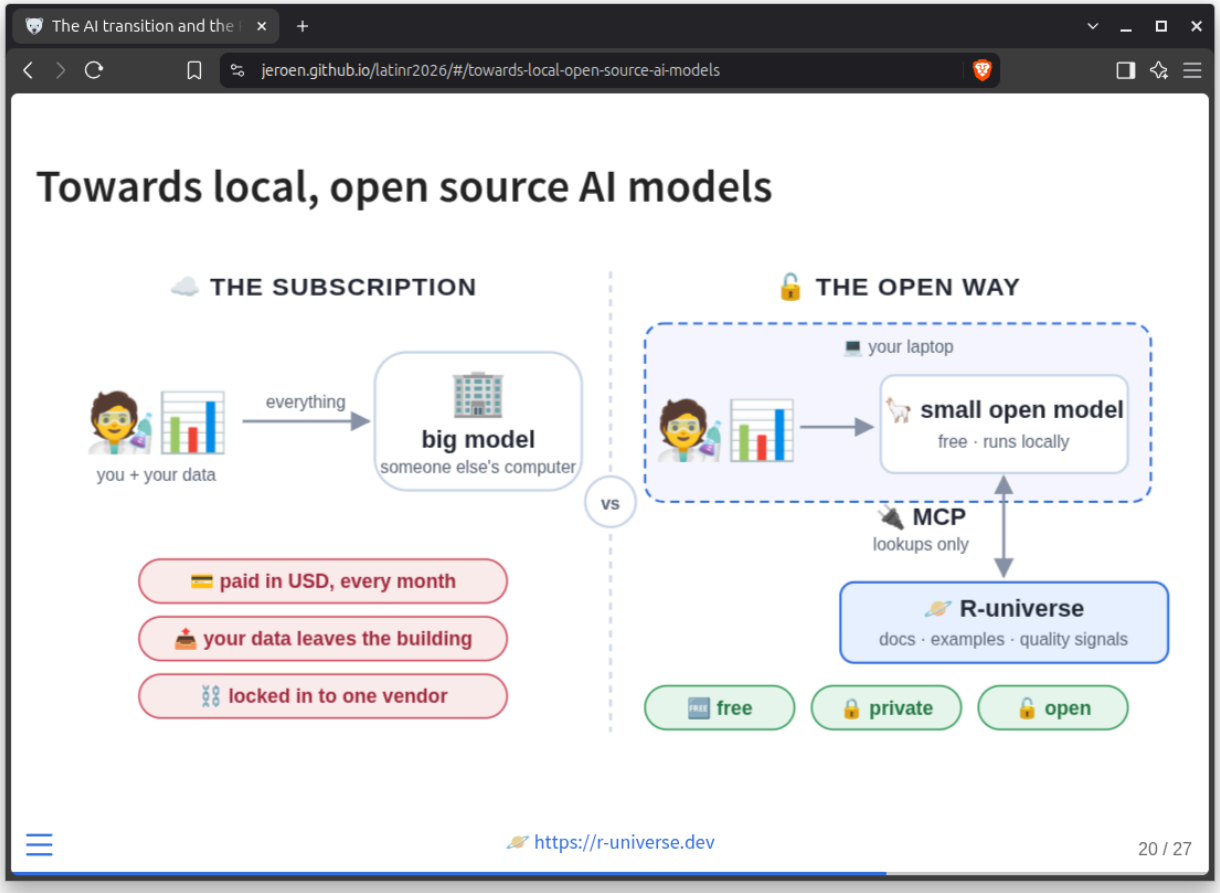
ILLINOIS

This talk demonstrated a few things:

- A working setup on medium-sized hardware
- Bigger will always be better
  - But local models can be “good enough” for a lot of tasks
- Local models are frequently said to be ‘six months behind frontier models’
  - But that is still miles ahead of what was expected years ago
- Local models will see more development (see next slide)
  - Likely a wise idea to invest time into local models and experimentation
  - Software development has changed. Local models are compelling.

Jeroen last week at LatinR

ILLINOIS



ILLINOIS

Thank you!

This version last updated on 2026-11-18. The repository with support material is at https://github.com/eddelbuettel/talk-programming-with-local-models.