Package: clinChat (via r-universe)

November 12, 2026

Type Package

Title Conversational Interfaces for Clinical Data with Large Language Models

Version 0.1.0

Date 2026-10-22

Description Provides tools and Shiny applications for interacting with clinical datasets through natural language using large language models. Built on 'ellmer' and 'shinychat', it emphasizes privacy-aware workflows where data remains local and only necessary summaries or tool results are shared with the model. Includes ready-to-use chat interfaces for exploratory analysis, summarization, and querying of clinical data.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports shiny (>= 1.8.0), ellmer (>= 0.4.0), shinychat (>= 0.4.0), dplyr (>= 1.1.0), cli, rlang, glue

Suggests testthat (>= 3.0.0), knitr, rmarkdown, bslib

URL https://github.com/shekharchalwadigrepo-hub/clinChat

BugReports https://github.com/shekharchalwadigrepo-hub/clinChat/issues

VignetteBuilder knitr

RoxygenNote 7.3.2

NeedsCompilation no

Config/pak/sysreqs cmake make libuv1-dev libssl-dev zlib1g-dev

Repository https://shekharchalwadigrepo-hub.r-universe.dev

Date/Publication 2026-10-22 14:06:36 UTC

RemoteUrl https://github.com/shekharchalwadigrepo-hub/clinchat

RemoteRef HEAD

RemoteSha 94e560a7ea97bf89b720140f65e4f76b9a24f493

Contents

clinChat 2
clinical_tools 2
create_clinical_chat 3
run_clinchat_app 4

Index **5**

`clinChat` *Conversational Interfaces for Clinical Data with Large Language Models*

Description

Provides tools and Shiny applications for interacting with clinical datasets through natural language using large language models. Built on ellmer and shinychat, it emphasizes privacy-aware workflows where data remains local and only necessary summaries or tool results are shared with the model.

Author(s)

Shekhar Chalwadi <skrchalwadicg@gmail.com>

See Also

Useful links:

- https://github.com/shekharchalwadigrepo-hub/clinChat
- Report bugs at https://github.com/shekharchalwadigrepo-hub/clinChat/issues

---

`clinical_tools` *Generate a list of tools for clinical data exploration*

Description

Generate a list of tools for clinical data exploration.

Usage

```
clinical_tools(data)
```

Arguments

`data`	A data.frame.

Value

A list of ellmer tool objects.

`create_clinical_chat` *Create a clinical data chat object*

Description

Creates an ellmer::Chat object pre-configured with tools for exploring a clinical dataset. Data remains local; only tool results are shared with the model.

Usage

```
create_clinical_chat(data, system_prompt = NULL, provider = NULL, ...)
```

Arguments

`data`	A data.frame containing clinical data.
`system_prompt`	Character. System prompt for the model. Defaults to a clinical analyst persona that emphasizes accuracy and privacy.
`provider`	Character or NULL. Provider string for ellmer::chat(), e.g. "openai/gpt-4o-mini". If NULL, uses the default from ellmer.
`...`	Additional arguments passed to ellmer::chat().

Value

An ellmer Chat object with clinical tools registered.

Examples

```
# Create sample clinical data (executable)
data <- data.frame(
  patient_id = 1:20,
  age = sample(25:80, 20, replace = TRUE),
  sex = sample(c("M", "F"), 20, replace = TRUE),
  outcome = sample(c("responder", "non-responder"), 20, replace = TRUE)
)

# Inspect available tools (no API key required)
tools <- clinical_tools(data)
length(tools)

# Creating a chat object requires a configured LLM provider
if (FALSE) {
  chat <- create_clinical_chat(data, provider = "openai/gpt-4o-mini")
  # chat$chat("How many responders are there by sex?")
}
```

`run_clinchat_app` *Launch a Shiny chat application for clinical data*

Description

Starts an interactive Shiny app that lets users chat with a clinical dataset using a large language model. Uses shinychat for the interface and ellmer for the model backend.

Usage

```
run_clinchat_app(
  data,
  system_prompt = NULL,
  provider = NULL,
  title = "Clinical Data Chat",
  ...
)
```

Arguments

`data`	A data.frame of clinical data.
`system_prompt`	Optional custom system prompt.
`provider`	Provider string for ellmer (e.g. "openai/gpt-4o-mini").
`title`	Title shown in the app.
`...`	Additional arguments passed to shiny::runApp().

Value

Invisibly returns the Shiny app object. Called for its side effect of launching the app.

Examples

```
# Sample data
data <- iris[1:20, ]

# Launching the app requires a configured LLM provider
if (FALSE) {
  run_clinchat_app(data, title = "Iris Explorer")
}
```

Index

∗ **package**
clinChat, 2

clinChat, 2
clinChat-package *(*clinChat*)*, 2
clinical_tools, 2
create_clinical_chat, 3

run_clinchat_app, 4