



CIMS GLOBAL & PHUSE 2026

ML07: Integrating LLM using R Shiny for Clinical Data Review by Ensuring Data Privacy and Validity

Peng Zhang, Ziqian He, Zhen Wu

Bio

Peng Zhang, Ph.D.

Associate Director, Innovative Data Sciences

CIMS Global



Peng Zhang
pzhang@cims-global.com

- Specializes in adaptive trial design and statistical monitoring of clinical trials
- Leads internal development of R packages and R Shiny applications to support regulatory and data review workflows

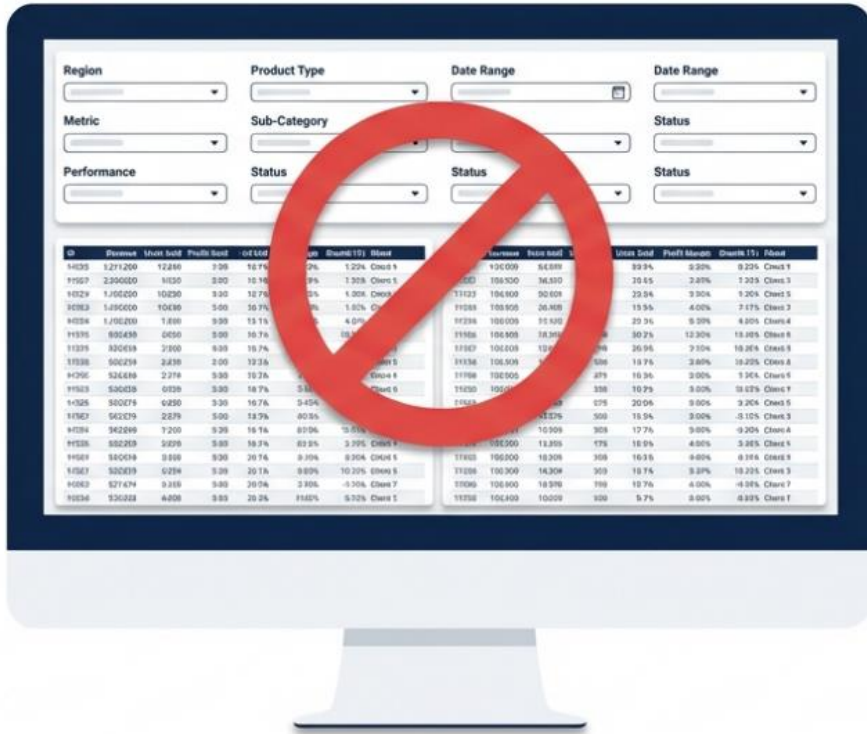
Previous Related Presentations

- *RISW 2025: Ensure the statistical validity in context of open-source approach – Peng Zhang, Christine Matakovich, Peilin Zhou, Tai Xie*
- *R/Pharma US 2025 Virtual Session: Integrating LLM using R Shiny for Clinical Data Review by Ensuring Data Privacy and Validity-Zhen Wu, Ziqian He, Peng Zhang [[here](#)]*

Agenda

- 01** Painpoint
- 02** Introduction to DataChat
- 03** Methodology
- 04** Future works
- 05** Q&A Session

Current Bottleneck: Dashboard Fatigue



- Rigidity: Fixed UI elements limit the questions you can ask



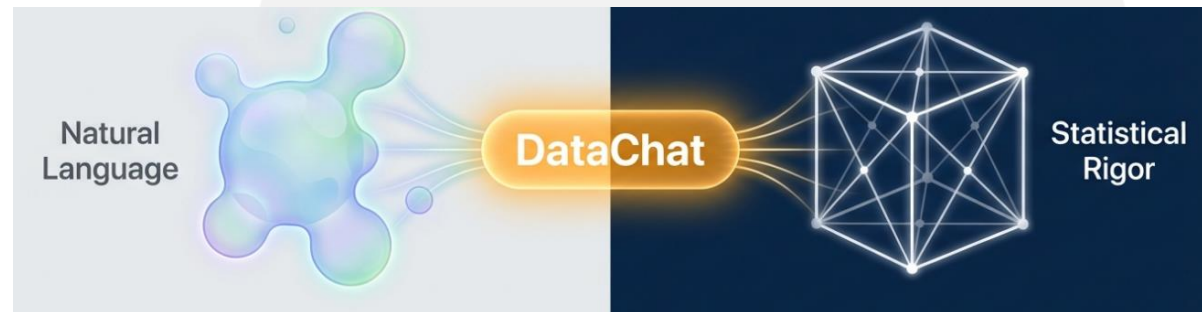
- Technical Barrier: Requires deep knowledge of variable names



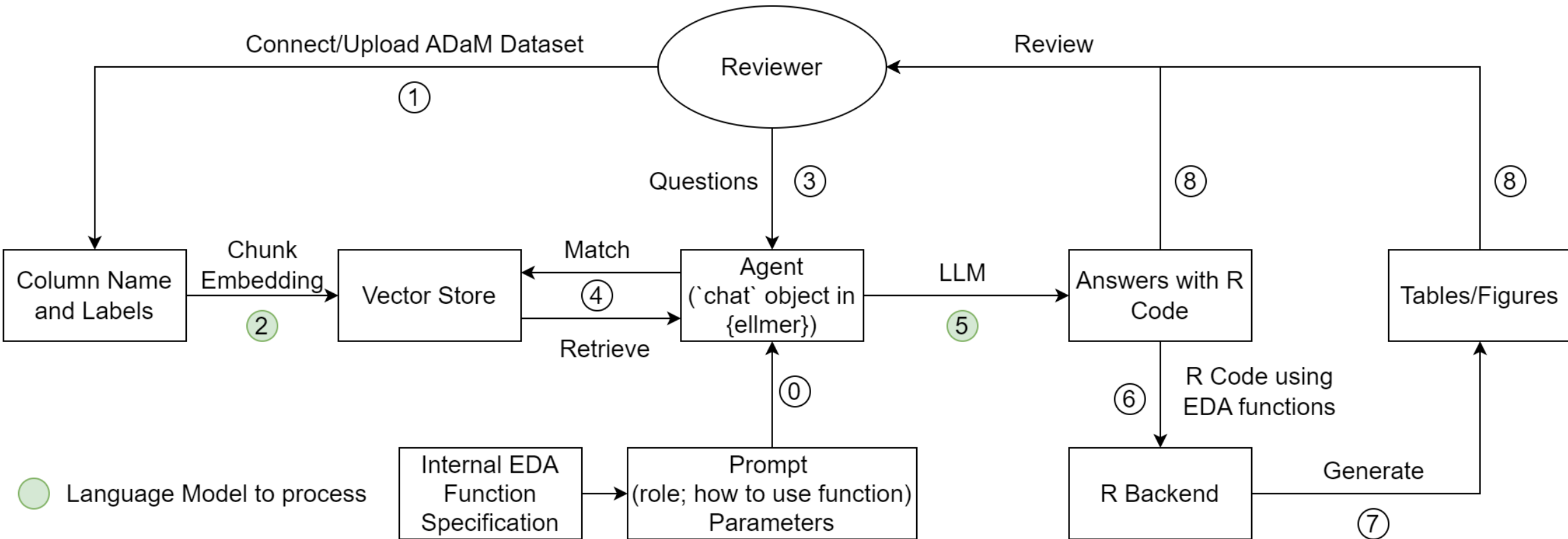
- The Lag: New questions = Weeks of programming delay

DataChat

- An R Shiny application that allows conversational interaction with clinical datasets
- Hybrid Approach: It bridges the gap between traditional dashboards and AI by adding a conversational layer on top of existing R workflows
- User Experience: Users ask questions in human language, and the system generates the corresponding analysis
- Guiding Principle: It is designed to complement existing reusable tools, providing an alternative interaction layer rather than a replacement



Backend Workflow



R Shiny Layout and LLM Features

Data Upload

⚠ Important: Please begin your chat after data is uploaded and columns are displayed below.

Connect ADaM data (accepts .sas7bdat or .zip with multiple .sas7bdat)

Browse... No file selected

Try out sample data!

Available Columns

{DataChat}: CDISC data query agent

Welcome to {DataChat}, your CDISC data query agent! I'm here to help you explore and analyze the data with tables and plots. Here are some examples of what you can ask me to do:

- "What is the age distribution by treatment group?"
- "Summarize ALT by treatment group."
- "Does age (ADSL) have association with adverse events (ADAE), given treatment (ADSL)?"
- "Is high baseline ALT associated with higher risk of Grade longer than 3 AE?"

Please upload your ADaM data above or use the sample data we provided.

Enter a message...

Visualization (guarantees data privacy)

Summary Table

Summary Plot

R Shiny Layout and LLM Features

Data Upload

⚠ Important: Please begin your chat after data is uploaded and columns are displayed below.

Connect ADaM data (accepts .sas7bdat or .zip with multiple .sas7bdat)

Browse... ADaM.zip

Upload complete

Try out sample data!

Available Columns

Dataset: ADLB

STUDYID (ADLB) : Study Identifier
USUBJID (ADLB) : Unique Subject Identifier
SUBJID (ADLB) : Subject Identifier for the Study
SITEID (ADLB) : Study Site Identifier
TRTP (ADLB) : Planned Treatment
TRTPN (ADLB) : Planned Treatment (N)
TRTA (ADLB) : Actual Treatment
TRTAN (ADLB) : Actual Treatment (N)
AVISIT (ADLB) : Analysis Visit
AVISITN (ADLB) : Analysis Visit (N)
ADY (ADLB) : Analysis Relative Day
PARAM (ADLB) : Parameter
PARAMCD (ADLB) : Parameter Code
PARAMN (ADLB) : Parameter (N)
AVAL (ADLB) : Analysis Value

{DataChat}: CDISC data query agent

Welcome to {DataChat}, your CDISC data query agent! I'm here to help you explore and analyze the data with tables and plots. Here are some examples of what you can ask me to do:

- "What is the age distribution by treatment group?"
- "Summarize ALT by treatment group."
- "Does age (ADSL) have association with adverse events (ADAE), given treatment (ADSL)?"
- "Is high baseline ALT associated with higher risk of Grade range than 3 AE?"

Please upload your ADaM data above or use the sample data we provided.

Enter a message...

Visualization (guarantees data privacy)

Summary Table

Summary Plot

R Shiny Layout and LLM Features

Data Upload

⚠ Important: Please begin your chat after data is uploaded and columns are displayed below.

Connect ADaM data (accepts .sas7bdat or .zip with multiple .sas7bdat)

Browse...

ADaM.zip

Upload complete

Try out sample data!

Available Columns

Dataset: ADLB

STUDYID (ADLB) : Study Identifier

USUBJID (ADLB) : Unique Subject Identifier

SUBJID (ADLB) : Subject Identifier for the Study

SITEID (ADLB) : Study Site Identifier

TRTP (ADLB) : Planned Treatment

TRTPN (ADLB) : Planned Treatment (N)

TRTA (ADLB) : Actual Treatment

TRTAN (ADLB) : Actual Treatment (N)

AVISIT (ADLB) : Analysis Visit

AVISITN (ADLB) : Analysis Visit (N)

ADY (ADLB) : Analysis Relative Day

PARAM (ADLB) : Parameter

PARAMCD (ADLB) : Parameter Code

PARAMN (ADLB) : Parameter (N)

AVAL (ADLB) : Analysis Value

{DataChat}: CDISC data query agent

Welcome to {DataChat}, your CDISC data query agent! I'm here to help you explore and analyze the data with tables and plots. Here are some examples of what you can ask me to do:

- "What is the age distribution by treatment group?"
- "Summarize ALT by treatment group."
- "Does age (ADSL) have association with adverse events (ADAE), given treatment (ADSL)?"
- "Is high baseline ALT associated with higher risk of Grade range than 3 AE?"

Please upload your ADaM data above or use the sample data we provided.

Enter a message...

Visualization (guarantees data privacy)

Summary Table

Summary Plot

{ellmer}

- {ellmer}¹ makes it easy to access the wealth of large language models (LLMs) from R.
- It allows user to use token and prompt to create Chatbots, extract structured data, and solve general programming problems.

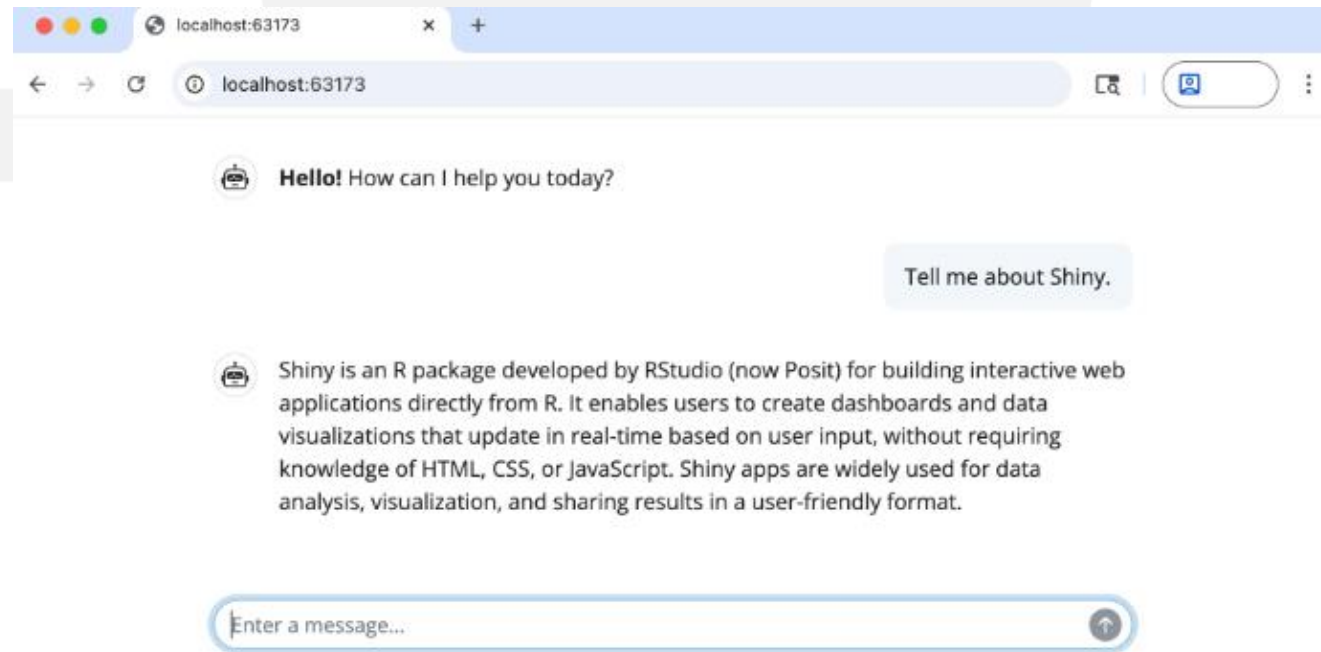
`live_console(chat)`

```
#> Entering chat console. Use "" for multi-line input.
#> Press Ctrl+C to quit.
#> >>> Who were the original creators of R?
#> R was originally created by Ross Ihaka and Robert Gentleman at the Unive
#> Auckland, New Zealand.
#>
#> >>> When was that?
#> R was initially released in 1995. Development began a few years prior to
#> in the early 1990s.
```

```
chat$chat("What preceding languages most influenced R?")
#> R was primarily influenced by S, which was developed at Bell Labs. Other
#> notable influences include Scheme, for its functional programming concept
#> various statistical programming languages like Fortran and Lisp.
```

{shinychat}

- {shinychat}¹ provides a Shiny toolkit for building generative AI applications like chatbots and streaming content.
- Users can create interactive and conversational experiences that make the data applications more engaging and user-friendly.
- It's designed to work alongside the {ellmer} package, which handles response generation.



Reactive Computation Logic & Backend Integration

Seamless Asynchronous Streaming with *ellmer* and *shinychat*



backend

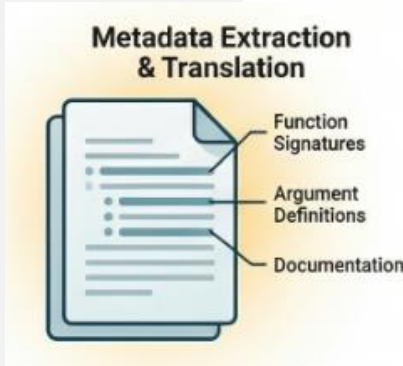
frontend



Tool Constrained Execution and UI Mapping

Step 1: Tool registration & R introspection

Registration via `chat$register_tool()`



ellmer uses R introspection to extract function signatures, argument definitions, and available documentation

Function/Component	Technical Role
<code>chat\$stream_async()</code>	Generates future/promise objects for non-blocking execution; releases R thread.
<code>chat_append()</code>	Handles SSE subscription and incremental DOM updates (chunk-based rendering).
<code>chat\$register_tool()</code>	Performs introspection on R functions to generate JSON Schema for agents.
R6 Class Object	Manages state persistence, context windows, and token serialization within the session.
shinychat UI	Implements Progressive Disclosure and multimodal rendering (htmltools/gt support).

Step 2: Constrained execution environment

Structured tool call generation
Local execution on shiny server
Security & data exposure prevention

Step 3: UI mapping & presentation layer

Tables and figures generated by R

{ragnar}

- {ragnar}¹ is an R package that helps implement Retrieval-Augmented Generation (RAG) workflows.
- It focuses on providing a complete solution with sensible defaults, while still giving the knowledgeable user precise control over each step.

Conceptual pipeline

- Metadata are explicitly authored in R
- Metadata are transformed into semantic embeddings
- An index is constructed for similarity-based retrieval
- Queries retrieve relevant metadata chunks

What ragnar guarantees

- Deterministic indexing given fixed metadata
- Bounded and inspectable retrieval results
- No access to raw data or numerical values

Why RAG?

- LLMs operate on text sequences; they do not seem to possess a concept of “facts” and “truth” like humans do.
- When the generated output is wrong, we call that a **hallucination**.
- RAG addresses this by retrieving relevant excerpts from a corpus of trusted, vetted sources and asking the LLM to summarize, paraphrase, or answer the user’s question using only that material.

Vocabulary

- It all starts with a **prompt**, which is the text (typically a question or a request) that you send to the LLM.
- Then we have a **conversation**, a sequence of turns that alternate between user prompts and model responses.
- Inside the model, both the prompt and response are represented by a sequence of **tokens**, which indicate either individual words or subcomponents of a word.
- The tokens are used to compute the cost of using a model and to measure the size of the **context**, the combination of the current prompt and any previous prompts and responses used to generate the next response.

Prompt

- The user prompt is the question that you send to the model.
- There are two other important prompts that underlie the user prompt:
 - **Platform prompt**: unchangeable, affects every conversation.
 - **System prompt**: set with new conversation, affects every response.

Conversation

- A conversation with an LLM takes place through a series of HTTP requests and responses
- In other words, a conversation consists of a sequence of a paired turns: a sent prompt and a returned response.



Privacy First Design:

What AI Sees

Subject ID	Age	Treatment	Outcome	Date	Status

Structure Only (Metadata)

What stays Local

John Doe	1975-05-12	MRN-12345	Placebo	Resolved	2023-11-01
John Doe	1975-05-13	MRN-12345	Placebo	Resolved	2023-11-01
Erivn Dams	1975-05-13	MRN-12345	Placebo	Resolved	2023-11-01
John Doe	1975-05-12	MRN-12345	Placebo	Resolved	2023-11-01
John Doe	1975-05-13	MRN-12345	Placebo	Resolved	2023-11-01
John Bobrin	1975-05-12	MRN-12345	Placebo	Resolved	2023-11-01
Edit Luck	1975-05-13	MRN-12345	Placebo	Resolved	2023-11-01
John Doe	1975-05-13	MRN-12345	Placebo	Resolved	2023-11-01
Emil Ron	1975-05-12	MRN-12345	Placebo	Resolved	2023-11-01
Ediuarson	1975-05-13	MRN-12345	Placebo	Resolved	2023-11-01
John Haunt	1975-05-13	MRN-12345	Placebo	Resolved	2023-11-01

Content (Subject Data)

- Retrieval is based on dataset metadata, not subject level clinical records
- Restricting data exposure to the LLM reduces privacy risks and supports regulatory compliance
- Computations are performed locally within the R session, keeping data secure

R Shiny Layout and LLM Features

Data Upload

Important: Please begin your chat after data is uploaded and columns are displayed below.

Connect ADaM data (accepts .sas7bdat or .zip with multiple .sas7bdat)

Browse... adsl.sas7bdat

Upload complete

Try out sample data!

Available Columns

STUDYID (ADSL) : Study Identifier

USUBJID (ADSL) : Unique Subject Identifier

SUBJID (ADSL) : Subject Identifier for the Study

SITEID (ADSL) : Study Site Identifier

ARMCD (ADSL) : Planned Arm Code

ARM (ADSL) : Description of Planned Arm

ARMCDN (ADSL) : Planned Arm Code (N)

TRT01P (ADSL) : Planned Treatment for Period 01

TRT01PN (ADSL) : Planned Treatment for Period 01 (N)

BRTHDTC (ADSL) : Date of Birth

RANDDT (ADSL) : Date of Randomization

ICDT (ADSL) : Date of Informed Consent

{DataChat}: CDISC data query agent

Welcome to {DataChat}, your CDISC data query agent! I'm here to help you explore and analyze the data with tables and plots. Here are some examples of what you can ask me to do:

- "What is the age distribution by treatment group?"
- "Summarize ALT by treatment group."
- "Does age (ADSL) have association with adverse events (ADAE), given treatment (ADSL)?"
- "Is high baseline ALT associated with higher risk of Grade range than 3 AE?"

Please upload your ADaM data above or use the sample data we provided.

summarize age by treatment

I found the following relevant variables in the dataset:

- **Age Variable:**
 - AGE (from ADSL): Age
- **Treatment Group Variable:**
 - TRT01P (from ADSL): Planned Treatment for Period 01

Now, I will generate the R code to summarize age by treatment using these variables.

```
cimsedapg_enhanced(  
  adam_data[['adsl']],  
  anal_var = "AGE",  
  group_var = "TRT01P",  
  dataset_name = "adsl"  
)
```

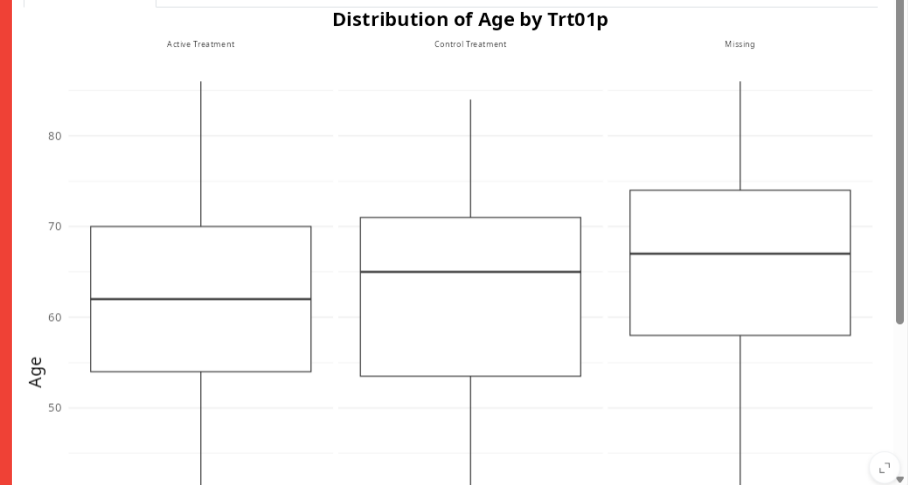
Enter a message...

Visualization (guarantees data privacy)

Summary Table

Characteristic	Active Treatment N = 173	Control Treatment N = 147	Missing N = 95
Age			
No. obs.	173	147	95
Mean (SD)	61 (11)	62 (12)	65 (12)
Median (Q1, Q3)	62 (54, 70)	65 (53, 71)	67 (58, 74)
Min, Max	26, 86	22, 84	27, 86

Summary Plot



Quick Demo (Video)

Data Upload


⚠ Important: Please begin your chat after data is uploaded and columns are displayed below.

Connect ADaM data (accepts .sas7bdat or .zip with multiple .sas7bdat)

No file selected

Available Columns

{DataChat}: CDISC data query agent

 Welcome to {DataChat}, your CDISC data query agent! I'm here to help you explore and analyze the data with tables and plots. Here are some examples of what you can ask me to do:

- "What is the age distribution by treatment group?"
- "Summarize ALT by treatment group."
- "Does age (ADSL) have association with adverse events (ADAE), given treatment (ADSL)?"
- "Is high baseline ALT associated with higher risk of Grade range than 3 AE?"

Please upload your ADaM data above or use the sample data we provided.

Visualization (guarantees data privacy)



UP NEXT

Implementation & Future

Discussion

- Small Language Model (local) vs Large Language Model (cloud API)
- Trade-off between retrieve accuracy and model efficiency
- Back-and-forth conversation stream

Conclusion

- Integrating large language models (LLMs) with R Shiny provides a practical solution for **clinical data review**.
- Our {DataChat} application leverages {ellmer}, {shinychat}, and {ragnar} to deliver a conversational interface that enables users to “**chat with data**” intuitively.
- {DataChat} empowers clinical reviewers without programming experience to explore, visualize, and interpret data within a **user-friendly** environment.
- The application ensures **data privacy** by keeping all computations local while sending only column metadata for embedding.
- It enhances result **accuracy** through retrieval-augmented generation (RAG), grounding responses in dataset structure and reducing hallucinations.

Contribution



Zhen Wu
zwu@cims-global.com

- Clinical Data Scientist at CIMS Global
- M.S. in Biostatistics, Yale University School of Public Health
- Experience of open-source development, including R packages, R Shiny, LLM engineering, Git, unit testing, and validation.



Ziqian He
zhe@cims-global.com

- Clinical Data Scientist at CIMS Global
- M.S. in Biostatistics, Columbia University School of Public Health
- Experience of open-source development and clinical trials from pre-clinical to phase 3.

Thank you!



©2024 CIMS Global. All rights reserved.