

Leveraging Gen AI for R Shiny App Code Development: A Case Study with Event Prediction

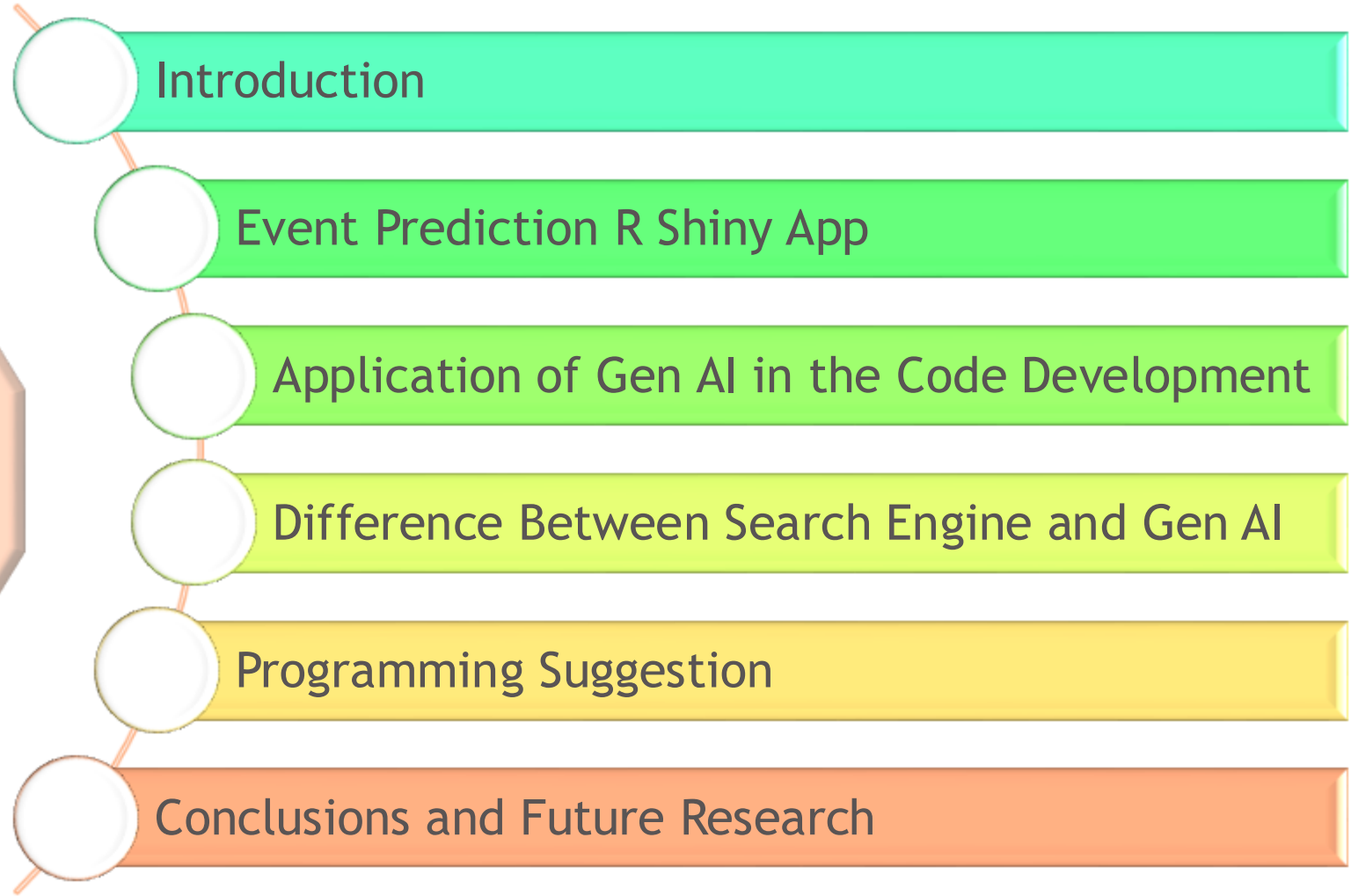
PHUSE EU Connect 2025 - Hamburg
Paper ML05

Yirong Cao
Director, Statistical Programming

Ding Jiang
Senior Manager, Data Science



Let's get started



Introduction



➤ Background:

- The development of Generative AI streamlines and automates the creation of R packages, resulting in faster prototyping, improved code quality, and robust validation processes.
- Implementing AI technologies can substantially accelerate the Software Development Lifecycle (SDLC), enhancing efficiency across all development phases.

➤ Objective:

This paper adopts a case study on R shiny apps demonstrates how to utilize AI to

- expedite repetitive tasks
- elevate the overall reliability and maintainability of the applications
- illustrate the vital role of AI in modernizing pharmaceutical code workflows

Event Prediction R Shiny App

- This R shiny app allows users to conduct simulation-based analyses, leveraging existing subject-specific data on randomization date, time to event, time to censoring, and dropout. This facilitates the prediction of analysis time for a target number of events.
- The application incorporates four tabs:
 - upload data
 - event prediction
 - summary tables
 - user manual



Application to Predict Analysis Times

Authors: Ding Jiang^a, David Paulucci^a, Diana Tai^b, Yirong Cao^b, Ming Zhou^c, Tianchen Xu^c, Cong Cao^c

[Upload Data](#)

[Event Prediction](#)

[Summary Tables](#)

[User Manual](#)

Results

Target Number of Events

50

Iterations of Simulation

1000

Assumed Distribution

- ☐ Exponential
- ☐ Exponential with Cure
- ☒ Piecewise Exponential
- ☐ Weibull
- ☐ Weibull with Cure
- ☐ Gompertz
- ☐ Gompertz with Cure

Time Points for Defining Breaks

- ☐ Define Actual Breaks
- ☒ Define Number of Time Points

The Number of Time-Points for Defining Breaks, e.g. 1, 2, 3 and 4

4

Enter Seed

1234

Confidence Level

95

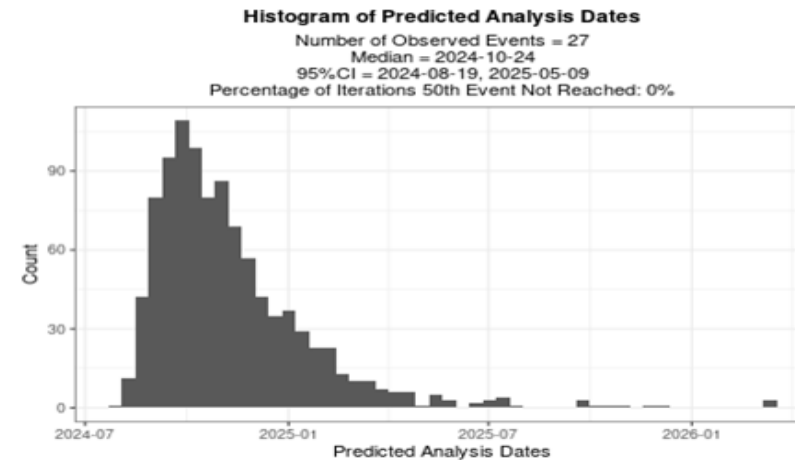
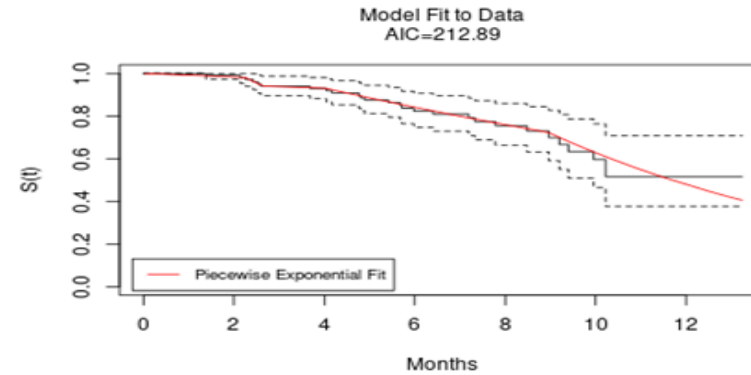
Cap Date

2025-05-06

Dropout Modeling

- ☒ Exponential
- ☐ Piecewise Exponential
- ☐ Manual (see previous page)
- ☐ Static

[Submit](#)



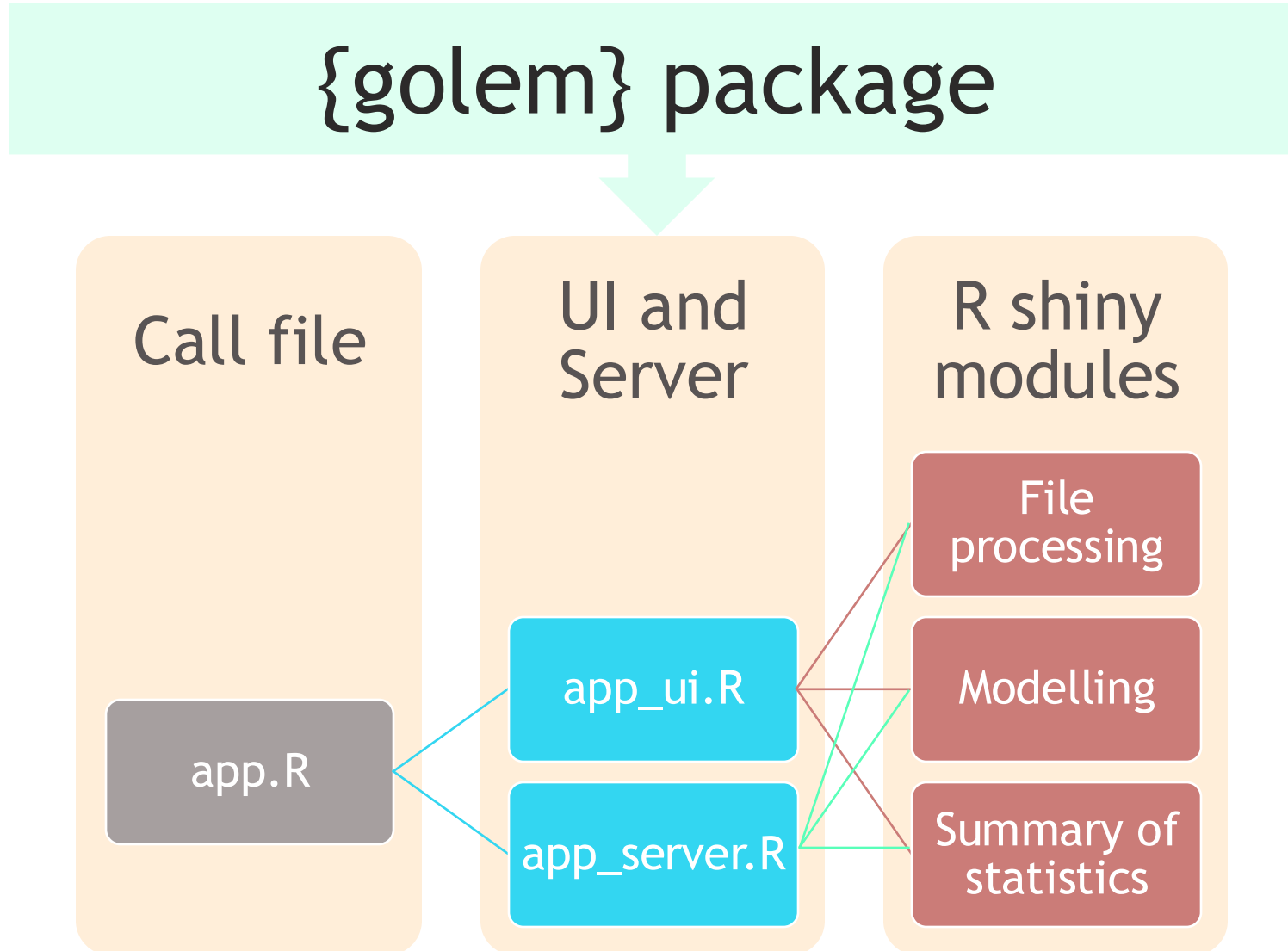
Probability that we will observe a target number of event 50 or more by 2025-05-06: 0.971

Events we will observe by 2025-05-06:

Median of the estimated events: 67

95% CI of the estimated events: 49, 81

Event Prediction R Package



Application of Gen AI in the Code Development

Learn Syntax

- Text with examples
- Immediate clarification via conversation

Create a New Example

- Tailored examples
- Options based on different R packages

Simplify Code

- Encapsulating reused code
- Standardizing modules

Debug

- Thorough explanation
- Correction

Difference Between Search Engine and Gen AI

Keywords vs Description

Search engines are highly optimized for concise, meaningful keywords. Their algorithms prioritize matching these keywords to indexed content across the web, maximizing relevance and search efficiency.

Gen AI models such as ChatGPT are built to process and understand detailed, context-rich prompts.

Quote vs Creation

Traditional search engines, such as Google, fundamentally operate by sourcing and quoting information directly from original resources found across the internet.

Gen AI models produce responses by summarizing, analysing, and generating content based on their extensive training data

Cutoff Date

Search engines are able to index and aggregate fresh content makes search engines significantly more effective for accessing the latest news, recent publications, and up-to-the-minute information across various fields.

ChatGPT's responses are informed by its most recent training data, which means that its knowledge is inherently limited by the cutoff date established during its last update.

Programmers Suggestion I: Code Design



Adopting modular approach

Submitting targeted queries for each module

Benefits:

- Maintainable and scalable structure
- Precise output

Programmers Suggestion II: Prompting



From Simple to Complex

Easy to debug and trace the error
Results are traceable



Be Descriptive

More details are returned
Accelerate the generation of actionable answers



Better to Give a Code Example

More accurate results
Be consistent

Example

- Step 1: Create an example of R shiny code with `radioButtons` and `updateRadioButtons`
- Step 2: Enhance this code to split into two R shiny modules: module 1 contains `radioButtons` and `updateRadioButtons`, and the values selected will be used in module 2 server

Programmers Suggestion III: Consistency

Items to be considered:

- Styles
- Structure
- Naming convention

Suggestions:

- Provide an example
- Detailed specification
- Documentation



Programmers Suggestion IV: Data Security

1

Sensitive
Information
Handling

2

Industry-
specific
Precautions

3

Use of
company-
trained AI tools



Conclusions and Future Research

- Key Take-away Points
 - Code development
 - Traditional search engine and Gen AI
 - Best practice
- Future research could incorporate Agentic AI to actively participate in code design processes, offering intelligent suggestions and autonomously evaluating multiple design options.



Any Questions?



Thank you

Your comments and questions are valued and encouraged. Please contact:

Yirong Cao
Bristol Myers Squibb
Uxbridge, UB8 1DH
United Kingdom
Yirong.Cao@bms.com

Ding Jiang
Bristol Myers Squibb
Lawrenceville
United States
Ding.Jiang@bms.com