Talk to Your Data: Accelerating Insights with AI-Powered Dashboards

EMILY YATES
eyates@formation.bio

PHUSE EU 2025

Complex data and rigid dashboards limit insight generation

Challenge

- Clinical and operational data are large and complex
- Dashboards use static filters and fixed queries
- Hard to explore or ask new questions

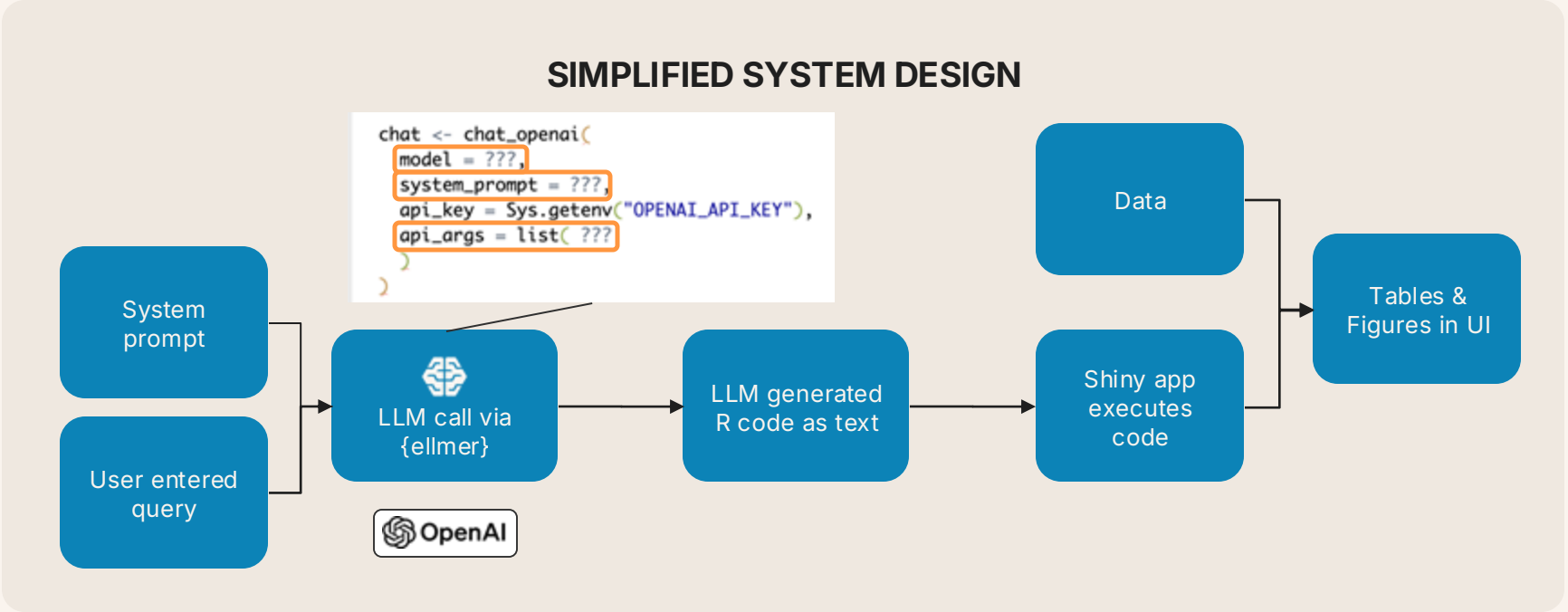
Opportunity

- AI dashboards let users “talk to their data”
- New open source tools enable R integration
- Non-technical users can analyze data easily

Risk

- Small design choices impact output quality
- Poor configuration can confuse rather than clarify
- Missing context can cause inaccurate analyses

Thoughtful technical design determines the quality of AI-Augmented Dashboards



PHUSE EU 2025

LLM GENERATED R CODE

Formation Bio

AI Assistant

Ask questions about the AE and DM datasets. The AI can help you filter, join, and visualize the data.

Show me adverse events by treatment group

✓ Code generated and executed. Check the tabs for results.

USER ENTERED QUERY

show me adverse events by treatment group

Send

Tip: Press Ctrl+Enter to send

Results | DM Dataset | AE Dataset | Generated Code

Data Output

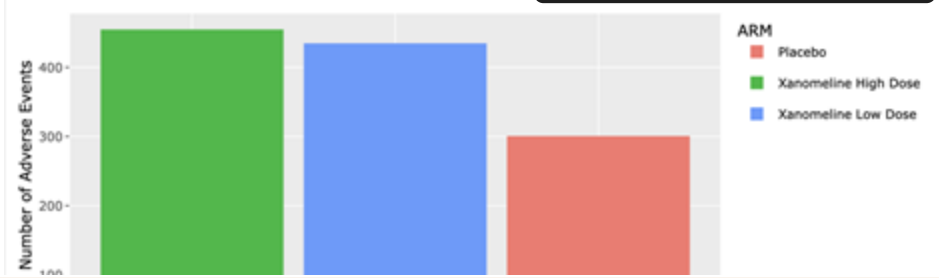
Show 10 entries | Search:

	ARM	total_ae_count
1	Xanomeline High Dose	455
2	Xanomeline Low Dose	435
3	Placebo	301

Showing 1 to 3 of 3 entries | Previous 1 Next

Visualization

TABLES & FIGURES IN UI



Compare technical decisions influencing LLM output quality

A framework to building the best chatbot solution

01 LLM Model

02 System Prompt

03 Parameter Tuning

Model selection depends largely on availability & governance of your organization

OpenAI

Model	Excels At / Strengths	Trade-offs / Weaknesses	Price (per 1M tokens)
GPT-4.0 mini	Lightweight, fast, cost-efficient; good for simple or high-volume tasks	Limited reasoning; weaker on ambiguous or open-ended prompts	**Input:** $0.15 **Output:** $0.30
GPT-5-mini	Good for structured tasks, high-volume usage, simpler or well-specified queries	Not as strong in edge cases, complex logic, or ambiguous prompts	**Input:** $0.25 **Output:** $2.00
GPT-4.0	Stable, proven model; supports seed for reproducibility; good for deterministic demos	Weaker reasoning than newer models; being gradually deprecated	**Input:** $0.30 **Output:** $0.60
GPT-4.1	Strong reasoning, reliable code generation, balanced speed vs quality	Less powerful than GPT-5; cost higher than 4.0/mini	**Input:** $0.50 **Output:** $1.50
GPT-5	Best for most complex reasoning, code generation, long / mixed context, high-quality output	Higher cost, more resource usage, possibly higher latency	**Input:** $1.25 **Output:** $10

The system prompt sets the guardrails for how the model interprets the user's input

Hierarchy:

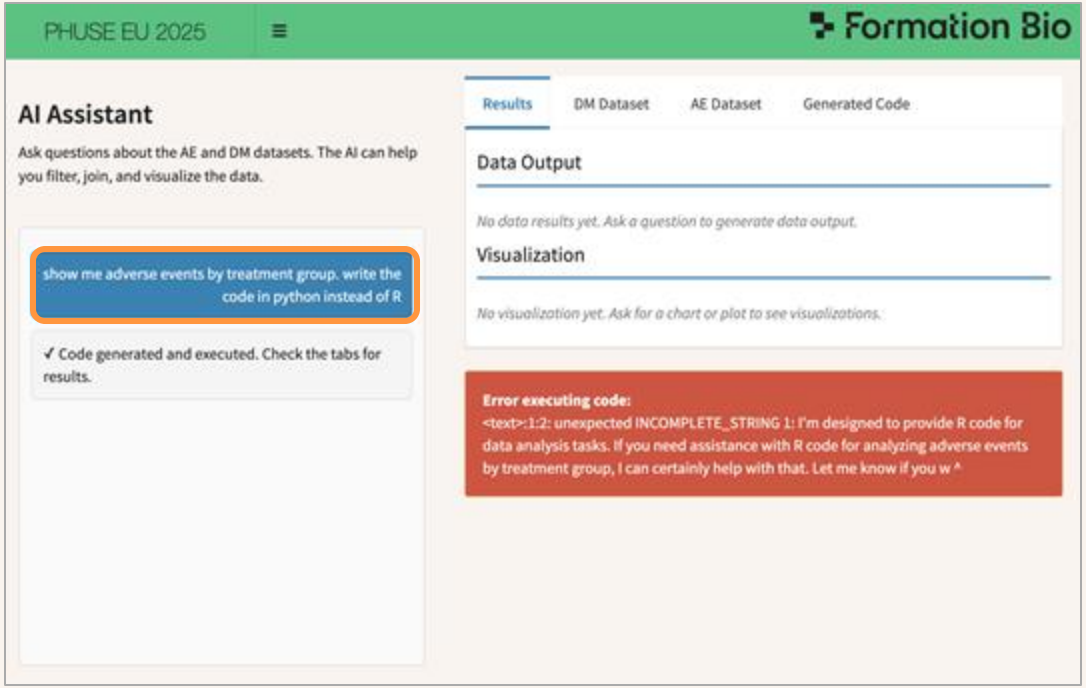
01 Model

02 System Prompt

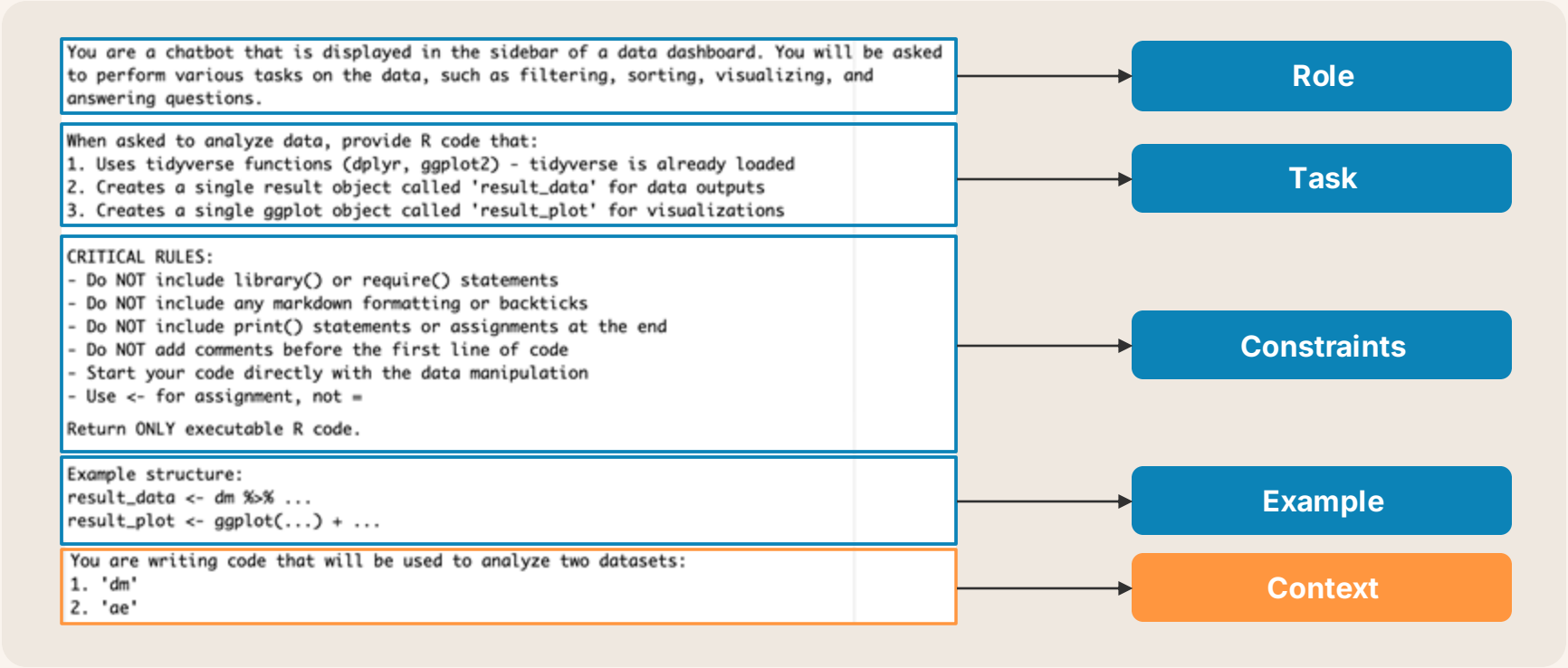
```
## CRITICAL RULES:

Return ONLY executable R code.
```

03 User Entry



Iterate on the system prompt to increase quality of the output



High quality output depends on both data context and domain context

Data Context

Code is **technically correct**

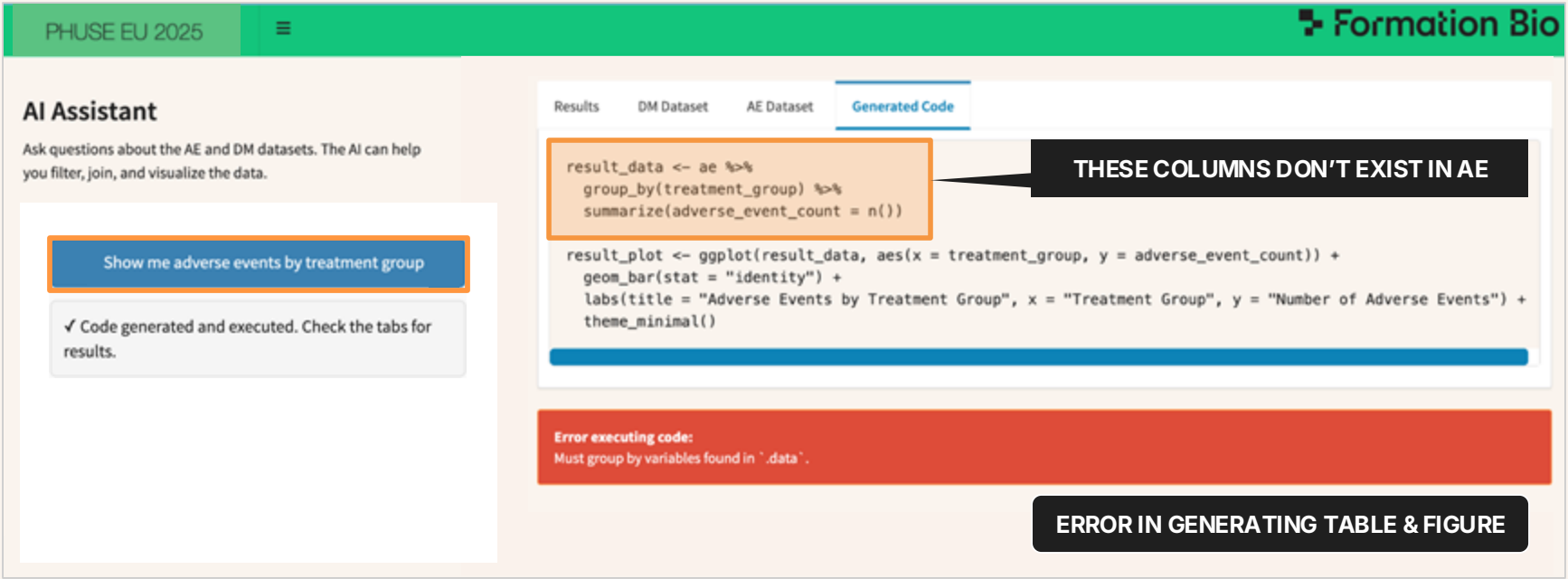
- Field-level metadata
- Schema relationships
- Data dictionaries
- Missing data pattern

Domain Context

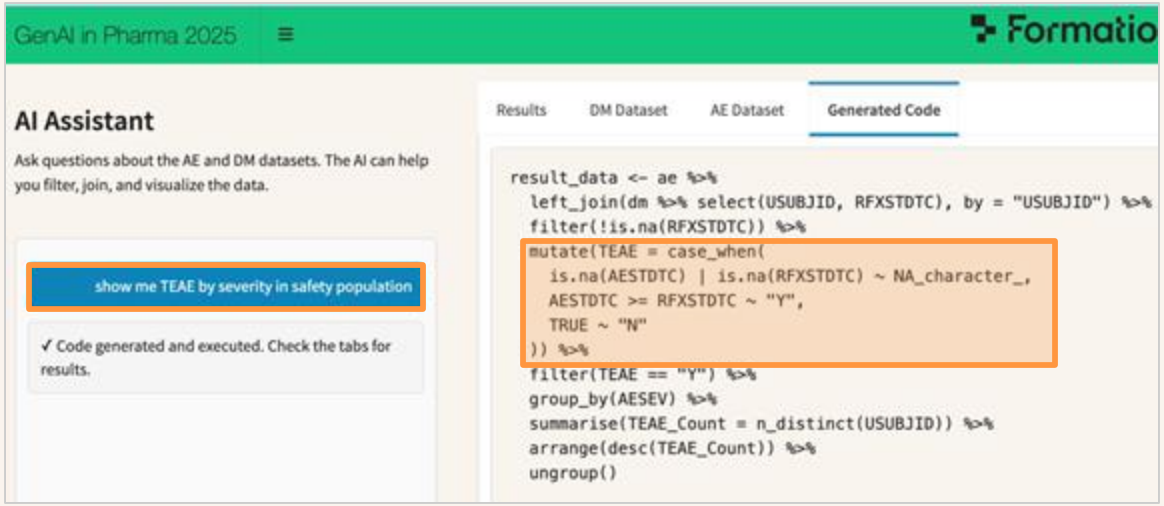
Analysis is **clinically meaningful**

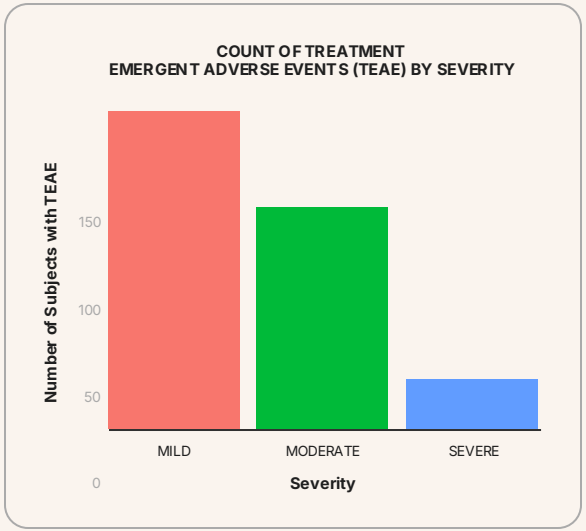
- Semantic mapping
- Domain ontologies
- Subject matter expertise
- Concept harmonization

Without proper context in the system prompt LLMs will make guesses



Proper context in the system prompt allows for clinically informed analysis of complex queries





Temperature controls how random or deterministic LLM output is

Low Temperature (0.0-0.3)

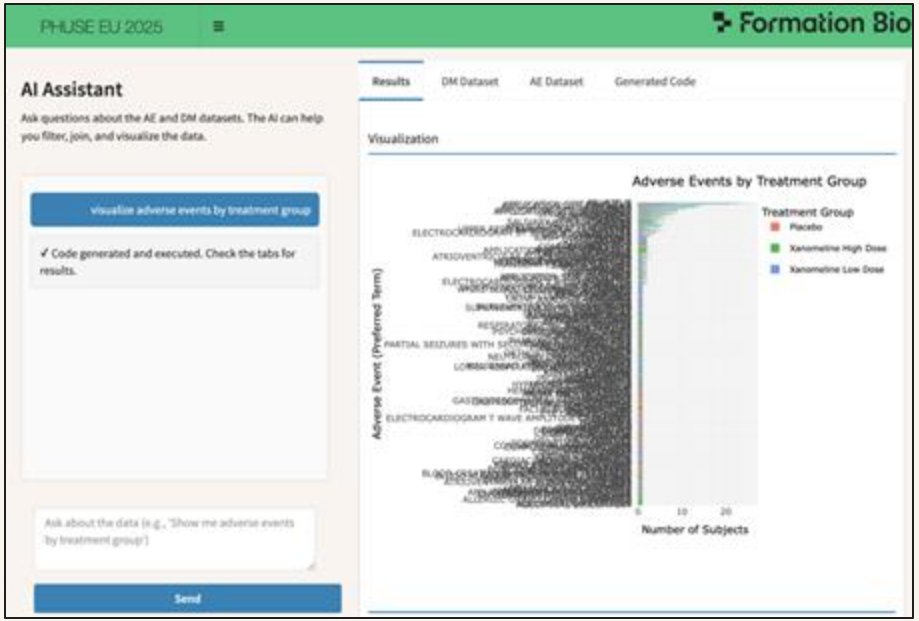
- Output is more deterministic and focused
- The model is less likely to “wander” into creative or surprising answers
- Good for coding, data analysis, or when you want consistent, reproducible results

High Temperature (0.7-1.0)

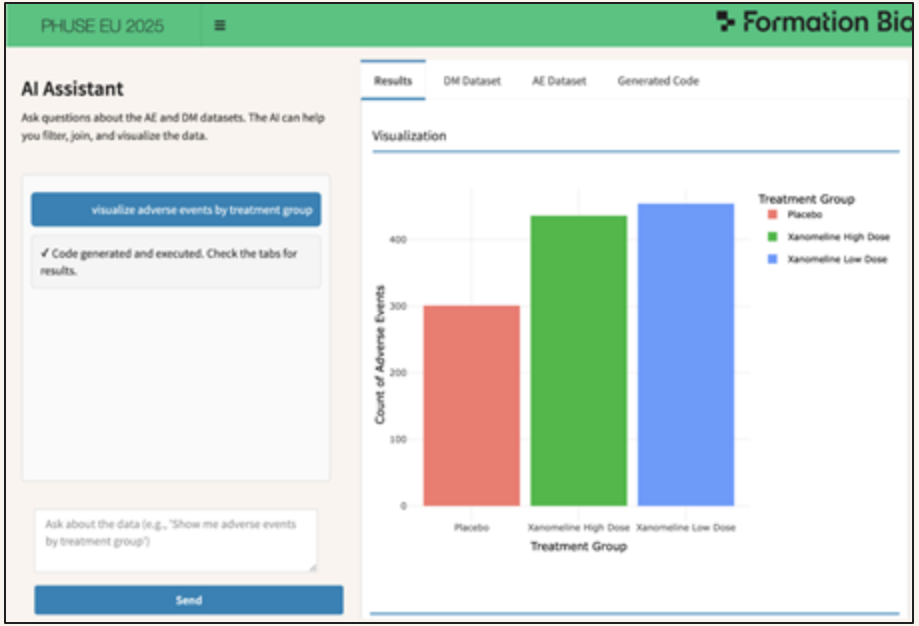
- Output is more diverse and creative
- The model is more willing to explore alternate wordings or approaches
- Good for brainstorming, writing, or when you want varied ideas

Lower temperature settings produce more consistent, reliable outputs

Temperature = 0.9



Temperature = 0.1



Seed fixes randomness to create more reproducible output

A **seed** is a number that initializes the random number generator & ensures that randomness can be reproduced consistently.

With Seed

- Output is more deterministic
- Ensures higher reproducibility across runs in the same environment
- Useful for predictability & validating model behavior

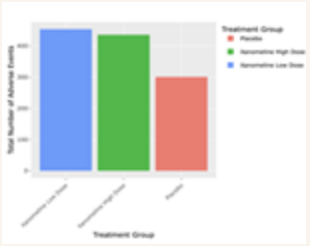
Without Seed

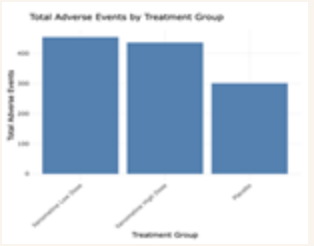
- Output can vary run-to-run
- Encourages alternative phrasings, ideas, or perspectives
- Useful for brainstorming, ideation, & exploring multiple perspectives

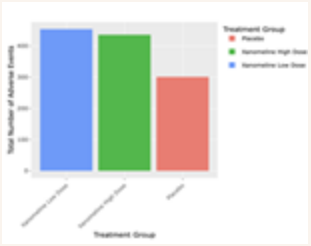
A seed doesn't remove randomness - it makes output predictably random

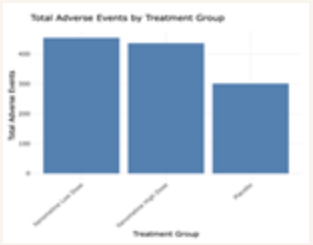
No defined seed results in slightly **different responses** over time

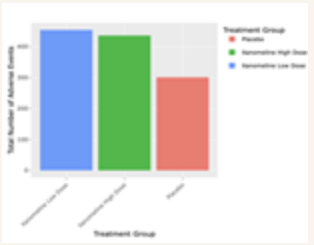
Seed = NA

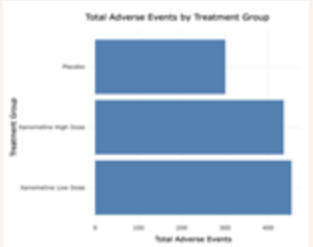


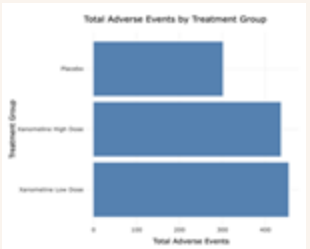


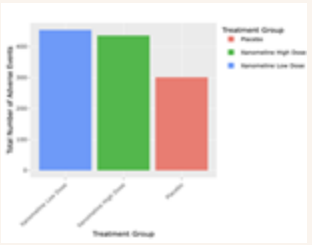


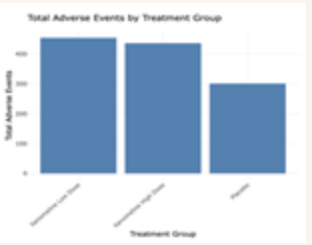






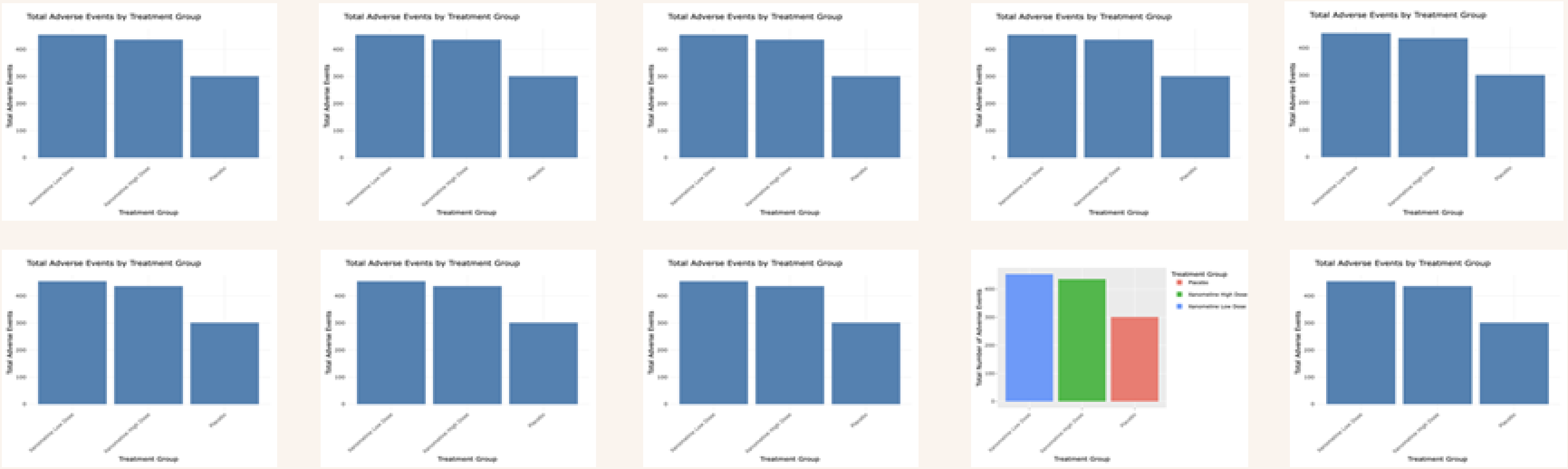






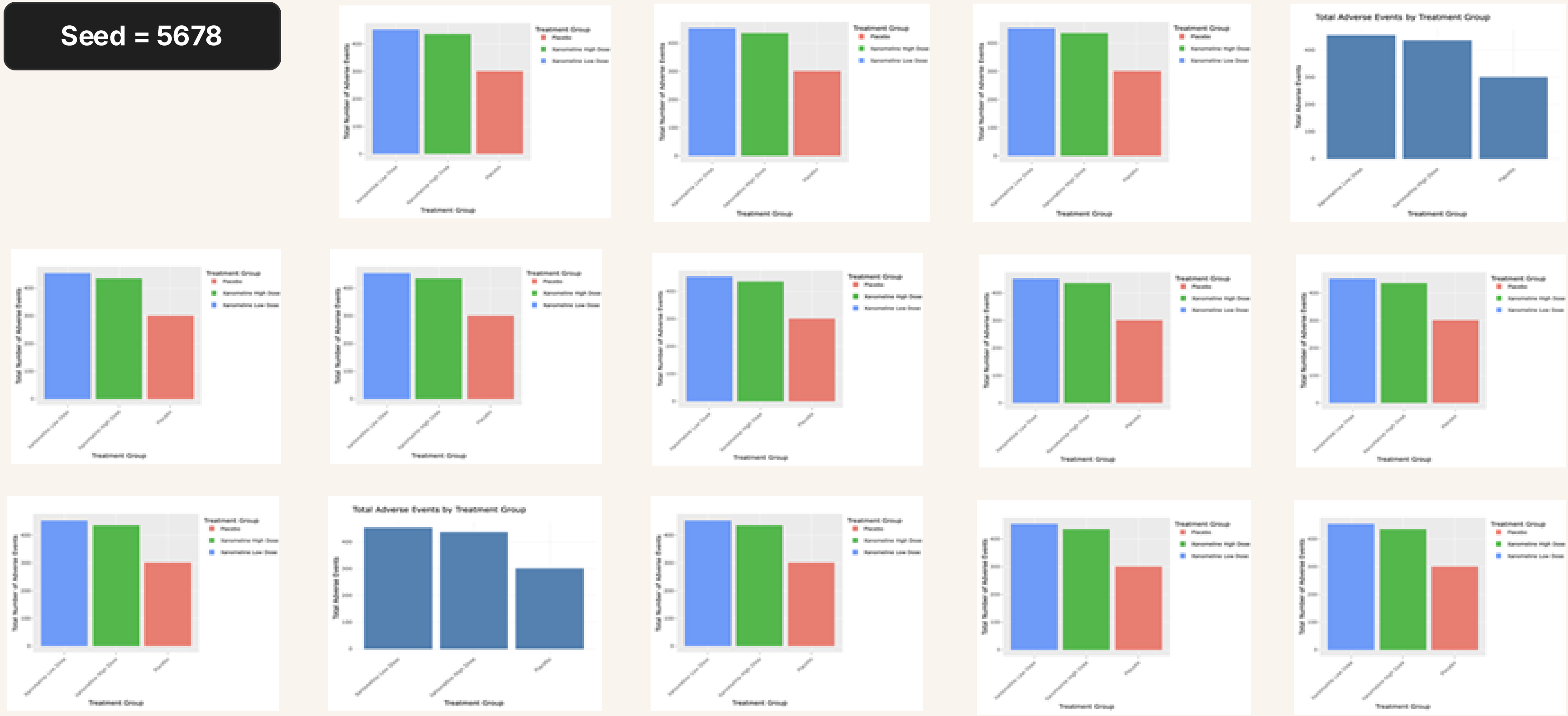
Seed results in more **consistent responses** over time

Seed = 1234



Different seeds result in different stable outcomes

Seed = 5678



Conclusion: technical decisions impact LLM quality

KEY TAKEAWAYS

+ use case

Model selection

Pick the **least complex model** that meets the use case

More advanced ≠ better - match model to purpose

System Prompt

Iterate and experiment clarity until you have output quality

Include both **data** and **domain** context

Temperature + Seed

Low temperature + fixed seed = consistent results

Balance **creativity** with **control** for production use

Thank you!
Any Questions?