Leveraging Julia for Advanced Data Visualization in Pharmaceutical Research

Ranjeet Kumar (ranjeet.kumar3@bms.com)
Deepak Vaishnav (deepak.vaishnav@bms.com)
Bristol Myers Squibb

Feb 19-21, 2026

Disclaimer

- This presentation is based on publicly available information (including data relating to non-Bristol Myers Squibb products or approaches).
- The views presented are the views of the presenter, not necessarily those of Bristol Myers Squibb.
- These slides are intended for educational purposes only and for the personal use of the audience. These slides are not intended for wider distribution outside the intended purpose without presenter approval.
- The content of this slide deck is accurate to the best of the presenter's knowledge at the time of presentation.

Agenda

1. The Importance of Data Visualization
2. Overview of the Current Visualization Landscape
3. Introduction to Julia
4. Visualization Tools within the Julia Ecosystem
5. Comparison of Performance Metrics
6. Regulatory Considerations
7. Where Julia fits today ?
8. Future Potential

Why Data Visualization Matters in Pharma

1

Complex clinical data

Clinical trials generate high-volume, high-dimensional data that requires advanced visualization to uncover patterns, trends, and anomalies efficiently.

2

Regulatory transparency

Clear and traceable visualizations enhance data interpretability, supporting regulatory review, audit readiness, and decision confidence.

3

Faster insights & decision-making

Interactive and responsive visualizations enable quicker insight generation, allowing teams to make timely, data-driven decisions.

Current Visualization Landscape

SAS, R, Python dominate

SAS, R, and Python form the backbone of current pharmaceutical analytics, covering regulated reporting, statistics, and exploratory visualization.

Performance & scalability challenges

As data volumes grow, performance limitations and scalability challenges become more visible in complex and interactive visualization workflows.

Growing data complexity

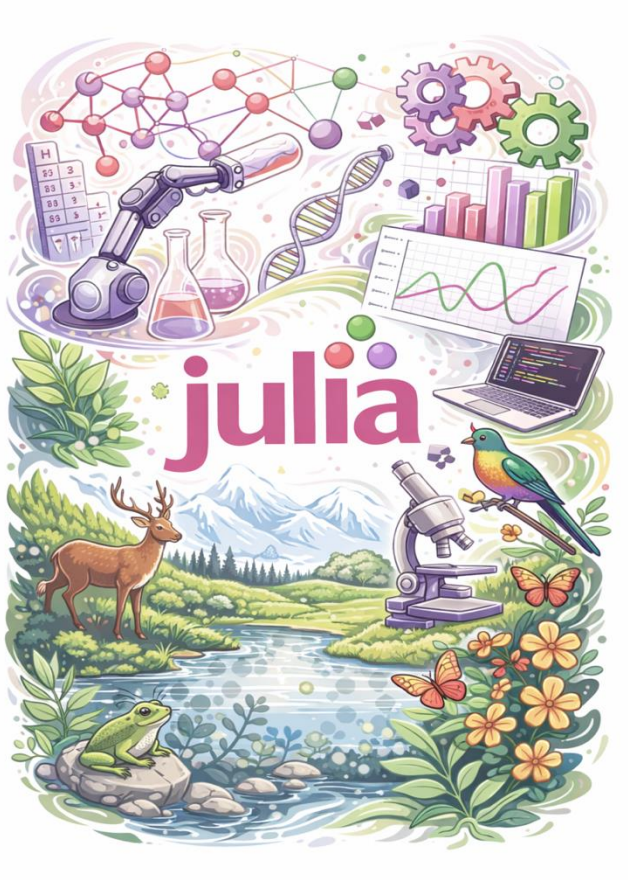
Modern clinical and operational datasets are increasingly high-dimensional, longitudinal, and heterogeneous, requiring more advanced visualization approaches

Introducing Julia

- High-performance language
- Designed for numerical & scientific computing
- Open-source

Why Julia Now?

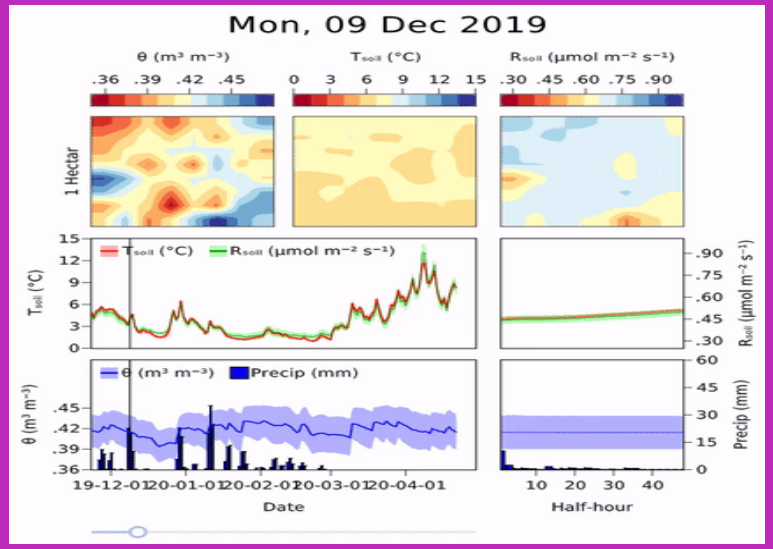
- Speed comparable to C
- Python-like simplicity
- R-like statistical strength



Julia Ecosystem for Visualization

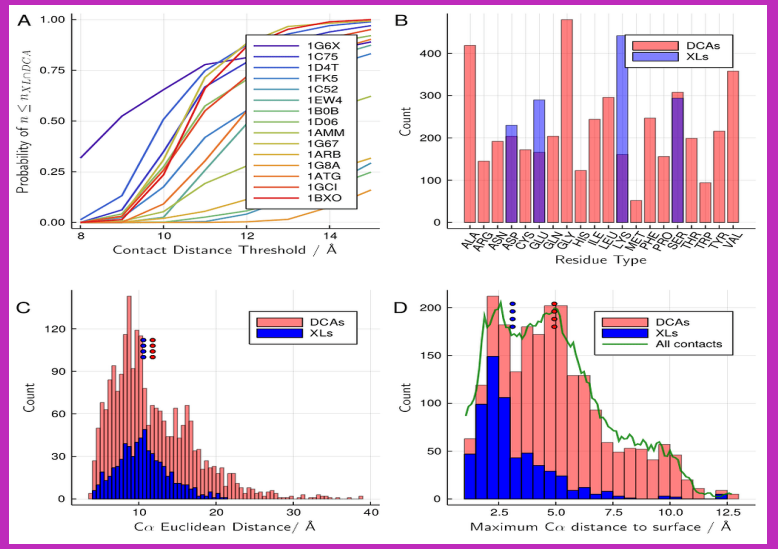
1 Makie.jl

Makie.jl enables high-performance, interactive, and GPU-accelerated visualizations for large and complex datasets.



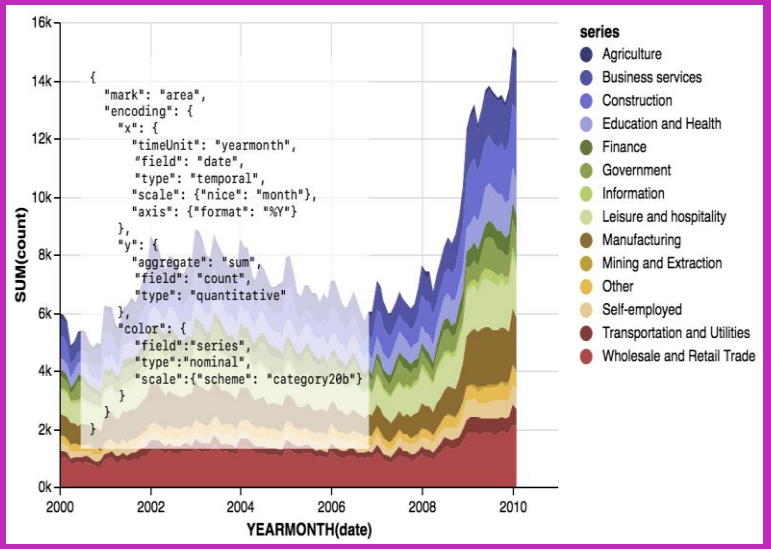
2 Plots.jl

Plots.jl provides a simple and flexible plotting interface that supports multiple backends for quick and consistent visualizations.



3 VegaLite.jl

VegaLite.jl enables declarative, grammar-of-graphics visualizations suitable for dashboards and exploratory analytics.

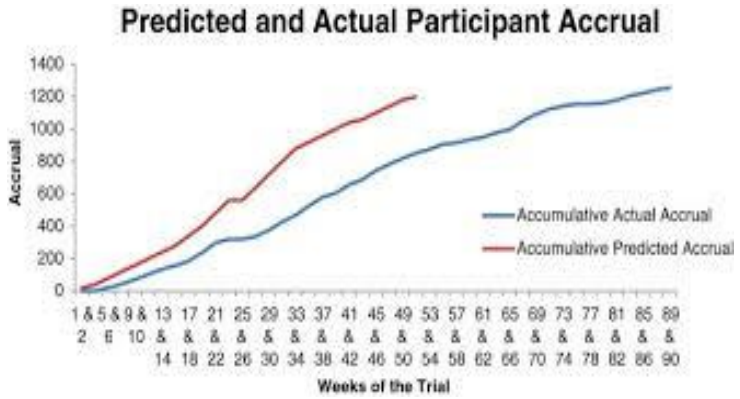


Clinical Trial Data Visualization

(from raw data to actionable insights)

- **Patient enrolment trends**

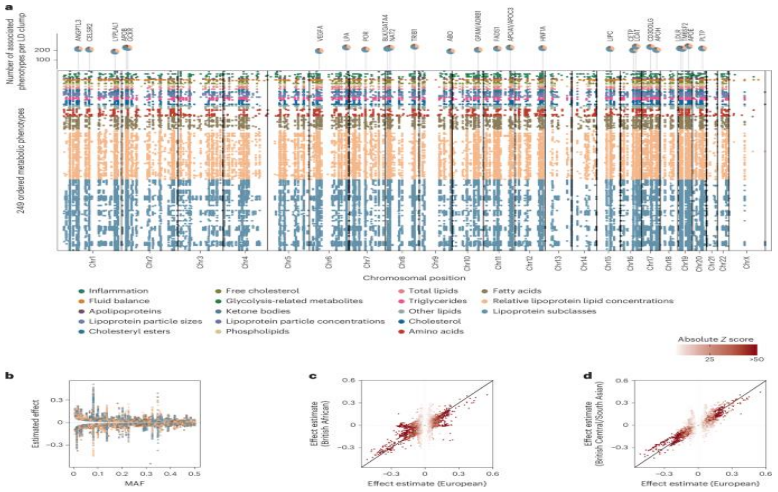
  - Early identification of **slow-enrolling sites**



Cumulative enrollment over time (line chart)

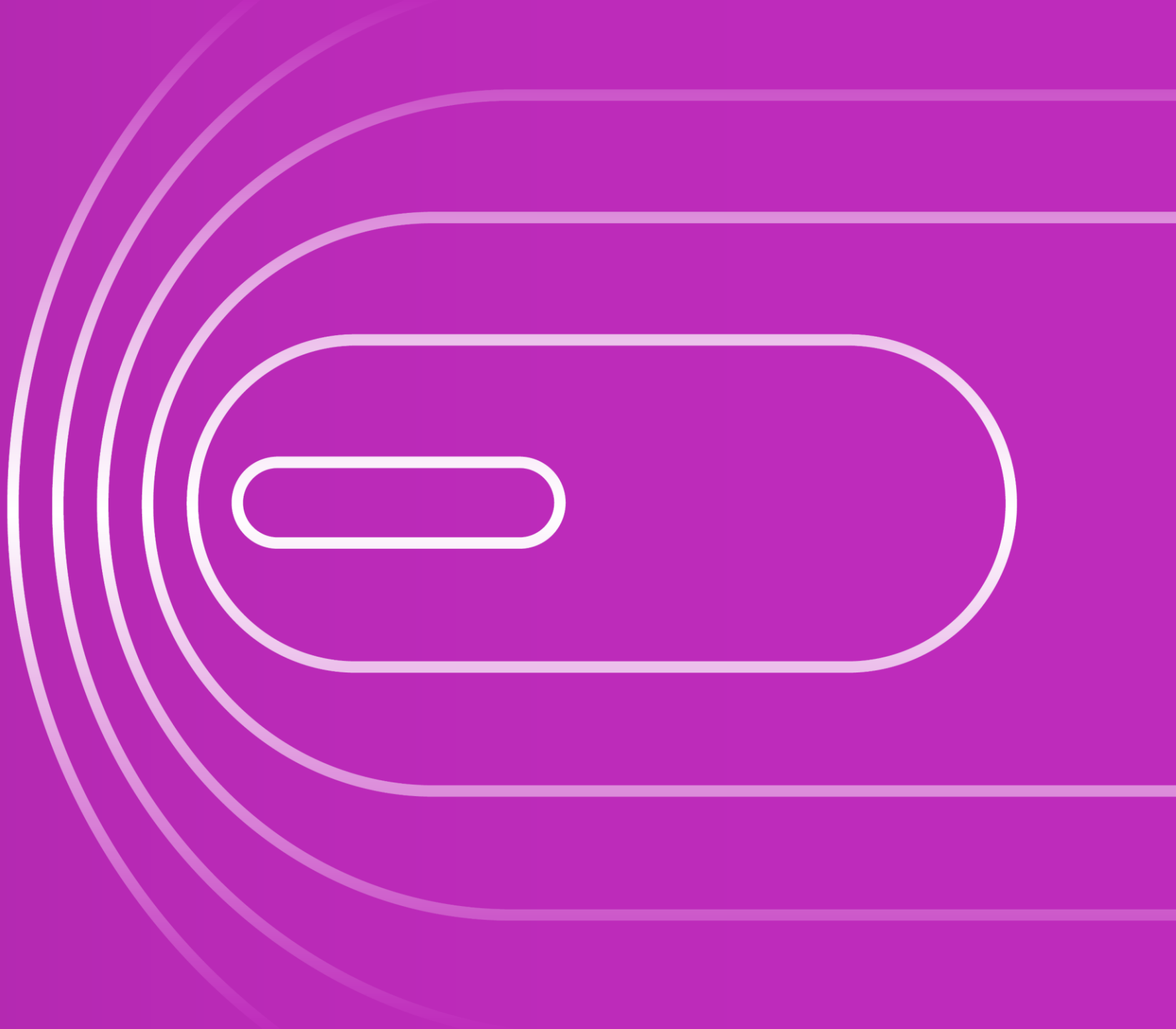
- **AE distributions**

  - Rapid drill-down into **high-risk events**



AE trends over time or by treatment arm

Performance Comparison



Julia Compared to R and Python: Performance & Interactivity

Julia vs R vs Python

- ✓ Julia delivers near C-level performance with high-level syntax
- ✓ R and Python often rely on optimized external libraries
- ✓ Julia avoids frequent language switching for performance tuning

Ex:
“In Julia, the same code used for prototyping can often be used for high-performance execution, without rewriting in C++.”

Large datasets

- ✓ Designed for efficient memory usage and parallel execution
- ✓ Handles millions of rows smoothly for analytics and visualization
- ✓ Performance scales as data size increases

Ex:
“Visualizing longitudinal lab data across thousands of subjects and visits without sampling or downscaling.”

Real-time interactivity

- ✓ Responsive zoom, pan, and filtering on large datasets
- ✓ Interactive dashboards update without long refresh delays
- ✓ Suitable for exploratory and operational monitoring

Ex:
“Real-time exploration of enrollment trends or AE distributions during ongoing trials.”

Pseudo Code Example

Load data

Transform

Visualize

Regulatory Considerations

GxP awareness

Validation strategy

Exploratory vs submission

Where Julia Fits Today

- Exploratory analytics
- Advanced visualization
- POCs

Industry Use Case:

- High-Performance Pharmacology Simulations
- Advanced Disease Modeling and Predictive Analytics

Future Potential

Hybrid workflows

Integration with Python/R

High-performance analytics

1. Start small
2. POCs
3. Training & governance

Adoption Strategy

Key Takeaways

Julia is promising

Not replacement but complement

Future-ready

Q & A ?

Thank you!

Reference

1. https://juliapackages.com/

2. https://github.com/