Paper DV05

Interactive Exploration of Top Line Results: Integrating DaVinci with Teal

Fabian Röthleithner, Boehringer Ingelheim Pharma GmbH & Co. KG, Biberach, Germany
Isabel Glauß, Boehringer Ingelheim Pharma GmbH & Co. KG, Biberach, Germany

ABSTRACT

Exploring top line results after database lock in clinical trials is crucial for quickly assessing effectiveness, safety, and statistical significance to evaluate success of a medication. Using modular R/Shiny apps for interactive data visualization enhances this process significantly by enabling dynamic and trial-specific analyses. This presentation introduces an approach for interactive exploration of top line results, based on a combined approach of DaVinci and Teal modules. DaVinci integrates seamlessly with Teal modules allowing for a quick and comprehensive visualization of top line results, facilitating storytelling and decision-making in the clinical trial process.

INTERACTIVE EXPLORATION OF TOP LINE RESULTS

In clinical development, the timely and accurate interpretation of top line results (TLRs) following database lock is critical. Traditionally, these results are shared in static PDF formats, which are often lengthy, dense, and difficult to navigate. This approach can slow down decision-making and increase the risk of misinterpretation or oversight. To address these challenges, we introduced an interactive exploration framework that leverages R/Shiny-based applications, specifically integrating DaVinci and Teal modules. This approach transforms the way TLRs are reviewed and communicated, offering dynamic filtering, drill-down capabilities, and tailored visualizations that support storytelling and facilitate faster, more informed decisions.

MODULAR APP CREATION

In a naive approach of using R/Shiny [1] for interactive data visualization, one can create a comprehensive stand-alone application, that satisfies the needs of as many studies as possible. It has been observed, however, that these stand-alone apps quickly become unwieldy and miss user expectations. Moreover, some minor trial-specific adjustments are impossible to be implemented without disrupting functionality for other trials. Therefore, instead of creating universal stand-alone apps, modular frameworks intend to provide individual modules. Such R/Shiny modules [2] serve as functional building blocks that can be easily linked together into customized apps. This customization enables trial teams to quickly build trial-specific but lean apps without extensive programming knowledge. Modular frameworks that follow this approach include DaVinci [3] and Teal [4].

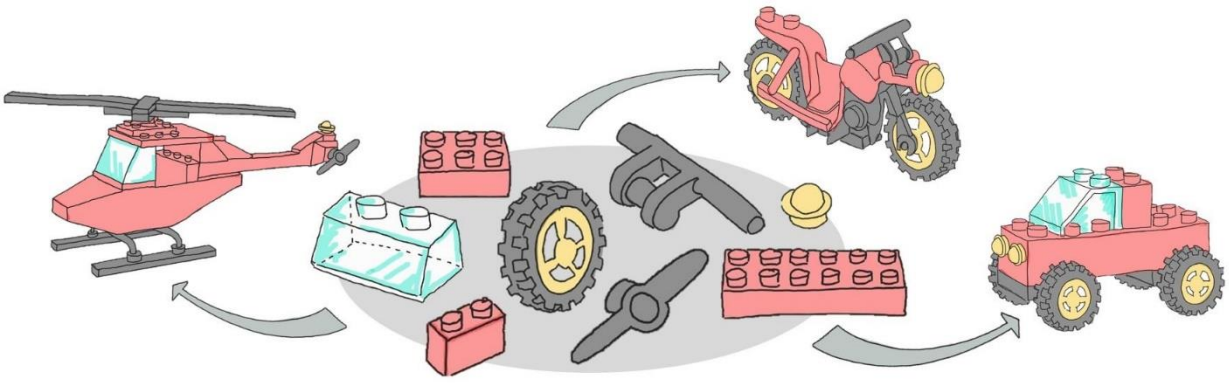

Figure 1: The concept of the modular approach explained by means of the analogy with LEGO bricks.

Figure 1 schematically demonstrates the modular approach by means of LEGO® bricks. In this analogy, the LEGO bricks symbolize the different modules. Depending on the use case, one might choose and assemble the LEGO bricks to build a helicopter, a motorcycle, or a pick-up truck. Similarly, the trial team will develop their trial-specific application by choosing and assembling the modules based on the team's unique requirements.

INTEGRATING DAVINCI AND TEAL

Both, DaVinci and Teal, are modular R/Shiny frameworks designed to support interactive clinical data exploration. Each offers a distinct set of modules tailored to different analytical needs. By using both frameworks, we aim to diversify the range of available modules, enabling trial teams to benefit from the strengths of each.

However, end users typically prefer a seamless experience and are reluctant to switch between separate applications. To address this, we implemented the {dv.teal} wrapper, which allows Teal modules to be embedded directly within DaVinci apps. This integration ensures that users can access a broader set of functionalities within a single, unified interface, enhancing usability and reducing cognitive load.

While some limitations remain, such as only partial support for bookmarking of Teal modules and the absence of Teal's reporter functionality, the combined framework significantly improves flexibility, scalability, and user satisfaction.

INSIDE THE TOP LINE RESULTS APP

With access to the module pools provided by both, DaVinci and Teal, we can assemble an output-rich application tailored to support TLR exploration. The resulting app includes a wide range of functionalities designed to meet the diverse needs of clinical trial teams:

- Visualization of lab values using boxplots, scatterplots, heatmaps, lineplots, and more (via DaVinci).
- Time-to-event and survival analysis with Kaplan-Meier plots, Cox regression, MMRM, and more (via Teal).
- Patient-level exploration and safety review, including patient profiles, AE hierarchy tables (DaVinci), demographic explorer, and variable browser (Teal).
- Utility modules like DaVinci's Bookmark Manager to enhance user experience.

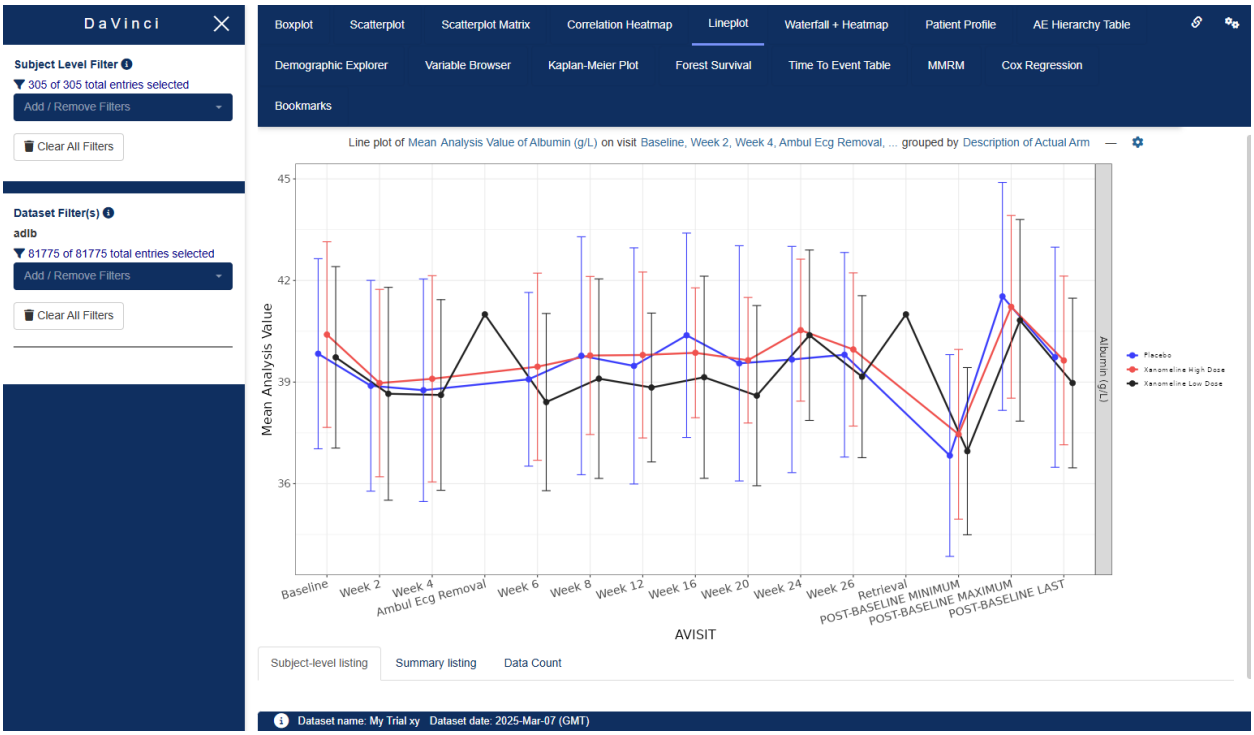

Figure 2a: Screenshot of a TLR application based on the DaVinci framework. Currently active output utilizes DaVinci's Lineplot module.

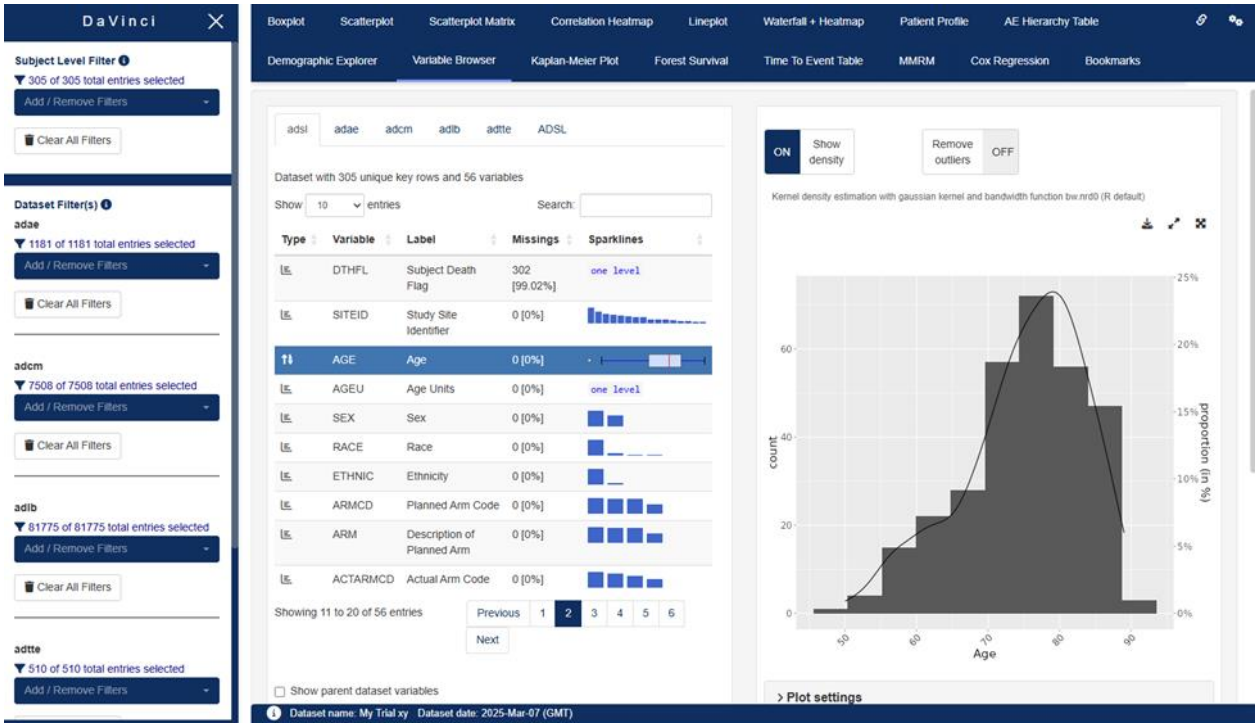

Figure 2b: Screenshot of a TLR application based on the DaVinci framework. Currently active output utilizes Teal's Variable Browser module.

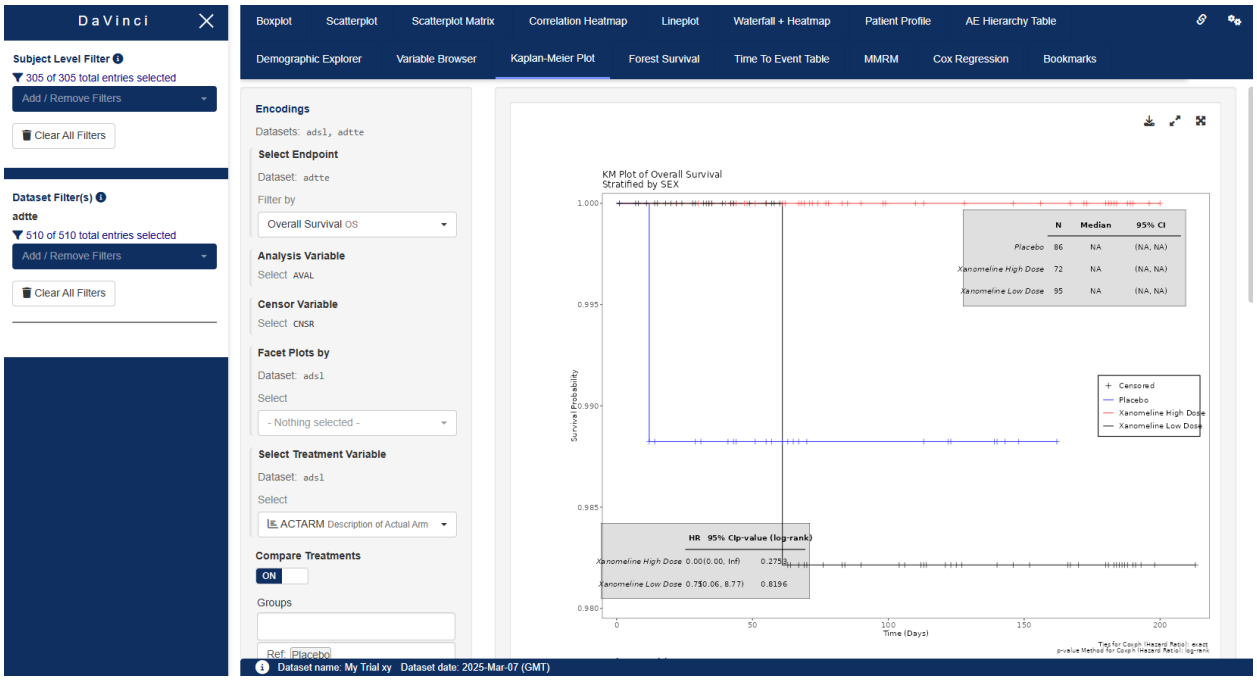

Figure 2c: Screenshot of a TLR application based on the DaVinci framework. Currently active output utilizes Teal's Kaplan-Meier Plot module.

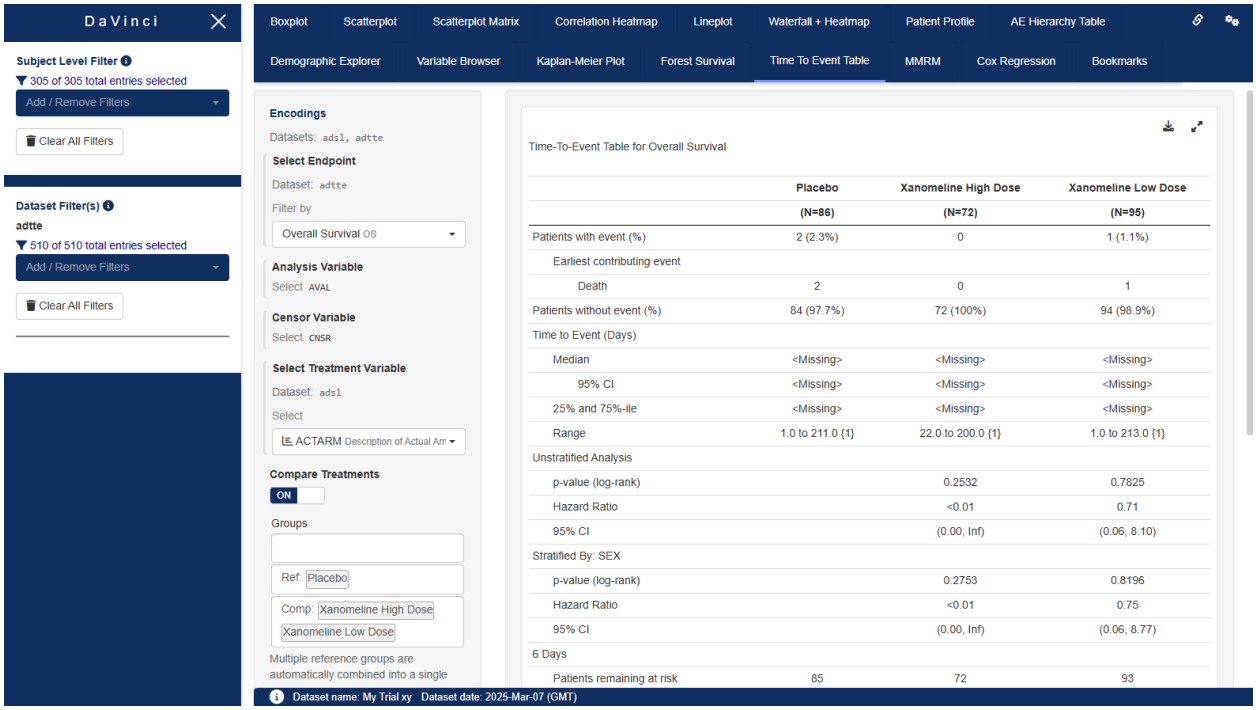

Figure 2d: Screenshot of a TLR application based on the DaVinci framework. Currently active output utilizes Teal's Time-to-Event Table module.

BRINGING TLR APPS TO THE TRIAL TEAMS

To ensure the value of interactive TLR exploration reaches clinical trial teams effectively, we developed a strategy focused on accessibility, adaptability, and scalability. At the heart of this approach is the Use Case Catalog, a curated collection of ready-to-use code templates. These templates serve as a practical starting point for trial teams, enabling them to quickly set up interactive apps tailored to their specific study needs. By minimizing the technical barrier, teams can focus on data interpretation rather than app development. The catalog supports a plug-and-play model: trial teams can select relevant modules, adapt them with minimal effort, and deploy them within their workflows. This empowers teams to explore TLRs interactively, enhancing their ability to identify key insights, communicate findings clearly, and make timely decisions. This initiative also fosters cross-functional collaboration, as teams from different departments can leverage a shared framework while customizing it to their unique requirements. Ultimately, bringing TLR apps to trial teams is not just about technology, it's about enabling better science through smarter tools.

OUR EXPERIENCES SO FAR

Implementing interactive tools for the exploration of TLRs has been a transformative journey. From a technical standpoint, the setup has proven to be robust and functional. The integration of modular R/Shiny components — particularly the combination of DaVinci and Teal — has enabled the creation of flexible, trial-specific applications that meet diverse study needs. However, success in this space goes beyond technology. One of the key learnings has been the importance of dedicated personnel. Developing, maintaining, and deploying interactive apps requires not only technical expertise but also a deep understanding of clinical data and user needs. Cross-functional collaboration has been essential. Feedback from various departments has been overwhelmingly positive. Users appreciate the clarity and effectiveness of visualizations, which significantly enhance communication and reduce ambiguity in interpreting results. These apps have also helped reduce workload, especially by minimizing the need for additional post-hoc analyses. That said, training and experience are crucial for optimal utilization. While the tools are powerful, their full potential is unlocked only when users are familiar with their capabilities and workflows. This has highlighted the need for ongoing support and education. Finally, we've learned that validation of modules benefits from a small, tailored approach. Rather than attempting to validate large, monolithic apps, focusing on individual minimal modules allows for more efficient testing and ensures reliability across different use cases.

REFERENCES

[1] Winston Chang, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert and Barbara Borges. *shiny: Web Application Framework for R.* R package version 1.9.1.9000, 2024. URL: https://github.com/rstudio/shiny, https://shiny.posit.co/.

[2] Hadley Wickham. Mastering Shiny. O'Reilly Media, Inc., 2021. ISBN: 149204735X, 9781492047353. URL: https://mastering-shiny.org/index.html.
[3] DaVinci homepage: https://boehringer-ingelheim.github.io/davinci/.
[4] Teal homepage: https://insightsengineering.github.io/teal/.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Fabian Röthleithner
Email: fabian.roethleithner@boehringer-ingelheim.com

Isabel Glauß
Email: isabel.glauss@boehringer-ingelheim.com

Boehringer Ingelheim Pharma GmbH & Co. KG
Birkendorfer Straße 65, 88397 Biberach an der Riß, Germany
Website: www.boehringer-ingelheim.com

Brand and product names are trademarks of their respective companies.