Paper OS08

The {teal} Framework Updates: How Open Source and Cross-Industry Collaboration Benefit Everyone!

Peyman Eshghi, Johnson and Johnson, High Wycombe, UK
Paul Jenkins, Johnson and Johnson, High Wycombe, UK

ABSTRACT

In today's data-driven landscape, effective data visualization is crucial for deriving insights and supporting informed decision-making. This paper presents recent advancements in the {teal} framework, an open-source R package that has gained significant adoption across pharmaceutical organizations for clinical data analysis and visualization.

We highlight the importance of data transparency and accessibility, especially in light of increasing regulatory interest, and discuss how the {teal} framework aligns with these expectations. Insights from industry leaders, gathered through forums like the PHUSE Working Group, provide context to our discussion. We present examples of applications of the {teal} framework, showcasing how Johnson & Johnson utilizes the framework and its new features to create interactive visualizations. Furthermore, we touch on the {teal} roadmap and future feature prioritization to illustrate ongoing developments. This paper aims to spark interest and stimulate conversation among participants regarding the potential of the {teal} framework in data visualization.

INTRODUCTION

The pharmaceutical industry faces unprecedented pressure to accelerate drug development while maintaining rigorous safety standards. The {teal} framework has emerged as a critical tool for interactive data visualization and analysis, providing a standardized platform for clinical data exploration that supports regulatory requirements.

This paper presents recent developments in the {teal} ecosystem, with particular emphasis on architectural enhancements that extend the framework's capabilities while preserving its core principles of modularity and reusability. We examine Johnson & Johnson's implementation experience, demonstrating how organizations can leverage transformators, decorators, and custom modules to address specific analytical requirements without compromising the framework's integrity.

Additionally, we discuss the strategic importance of cross-industry collaboration through initiatives such as the PHUSE Working Group, which has fostered the development of complementary packages like {uteals}. These collaborative efforts exemplify how open-source development can drive innovation while ensuring that solutions address common industry challenges.

Paper Structure: We begin by reviewing recent framework developments, then examine advanced components (transformators and decorators) with implementation examples, discuss enterprise access control through {rAccess} integration, and conclude with custom module development approaches for industry-specific requirements.

RECENT DEVELOPMENTS IN THE {TEAL} FRAMEWORK

- Improved the layout and appearance of the app using bslib components.
- General repositioning of key navigation components across the app:
    - **Modules**: The module navigation is moved from a nested tab selection to a "Module" drop-down selection. The module selection can be done from the nested button of modules.
    - **Reporter**: The Report previewer is no longer displayed as yet another teal module, it is placed inside a "Report" drop-down right next to the "Module" drop-down. The Report previewer is a modal that displays the added report cards. The "Report" drop-down also contains other global report options like download/load/reset the Report.
- **Teal Data Module**: The UI created by the teal_data_module() is now placed inside a modal that can only be closed when it returns teal_data.

ADVANCED FRAMEWORK COMPONENTS: TRANSFORMATORS AND DECORATORS

The introduction of transformators and decorators represents a significant architectural advancement in the {teal} framework, providing sophisticated mechanisms for customization while maintaining system integrity and performance.

TEAL TRANSFORMATOR

Transformators constitute reusable data processing components that operate at the data layer, enabling systematic modification and enhancement of datasets prior to module consumption. This architecture facilitates the implementation

of consistent data transformations across multiple analytical modules while eliminating code duplication and ensuring maintainability.

IMPLEMENTATION CASE STUDY: MULTI-VARIABLE OR FILTERING

In this example, we have added a transformator to support more robust querying requirements. Currently, the framework has the ability to use active filtering. However, this only supports 'OR' operations when the conditions are contained within one variable (for example, we can select grade 1 or grade 2 AEs by simply selecting those 2 options within the dropdown box for ADAE for variable AETOXGR). However, there is no current support for 'OR' operations when this spans multiple variables (for example, if we wanted to select all AEs related to Drug A or Drug B where relationship is captured in variables AEREL1 and AEREL2 respectively, this would not be possible without pre-processing a new variable for use within active filtering).

The logical filtering ('OR') transformator we created allows us to filter our dataset using the required 'OR' filtering with the addition of a simple line of code that can again be added to any existing teal module:

```
transformators = list(or_filtering_transformator("ADAE"))
```

We pass the dataframe into our transformator that we want the user to be able to use when doing their OR filtering within the specific module (we can pass multiple dataframes here if needed by adding another or_filtering_transformator() call to our list). The user will then see an additional UI panel in their teal module which will provide options for 'OR' filtering. They can build these queries, which provides a more flexible filtering mechanism than the active filter panel provides.

Figure 1.1

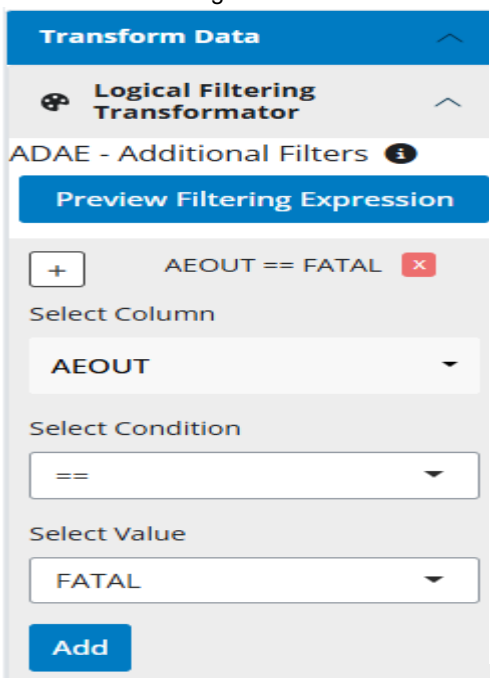
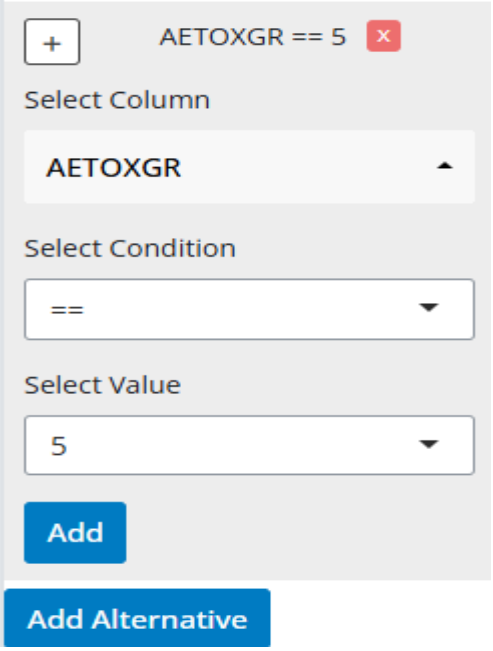
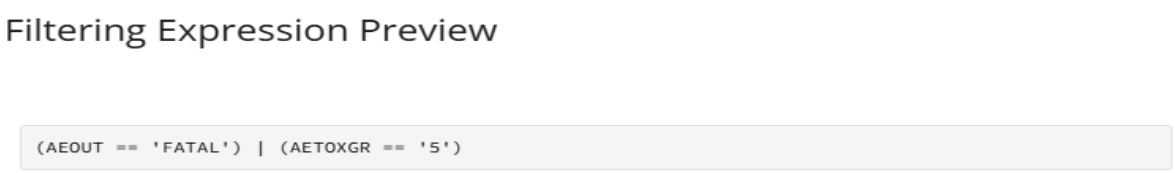


Figure 1.2



In this example, we have added an OR condition of AE with outcome of Fatal or Grade 5 AEs from the ADAE data frame. This can be seen by previewing the created expression.

Figure 1.3



TEAL DECORATOR

Decorators function as presentation layer components that enhance user interface elements and output formatting without altering core module functionality. This architectural pattern enables the systematic application of consistent styling, formatting, and UI enhancements across multiple analytical modules while maintaining separation of concerns.

EXAMPLE 1: DYNAMIC INSERTION OF TITLES AND FOOTNOTES

In this example, we were able to add a title and footnote decorator to our UI which connects us to a title/footnote file, enabling the user to either select an appropriate title from a list of pre-existing templates or to completely customize titles/footnotes. The code below can be added to any existing teal module call (for example, from a module within teal.modules.clinical).

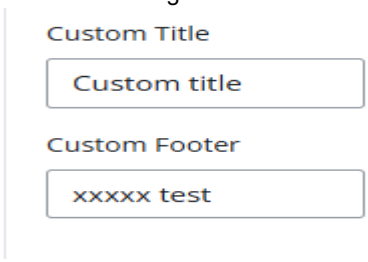
```
decorators = list(table = title_footer_decorator("table",
                 titles_file = "data/titles.xlsx"))
```

The screenshot below shows the UI that provides the user with a dropdown box for each tableID in the specified titles.xlsx file, which then inserts the relevant title taken from the file based on the tableID selected into the current teal module.

Figure 2.1

Furthermore, we have complete flexibility to customize this should we want something outside of the specified titles.xlsx file. Here we have added the custom title of “Custom Title” and the custom footnote of “xxxxx test” to the table instead of using a title from the titles.xlsx file.

Figure 2.2



EXAMPLE 2: GRAPH AXIS LABELS AND FONT SIZES

In this example, we have used a Johnson & Johnson developed teal module called tm_g_pp_remaining_j. We have added a decorator to this bar plot that can be used to decorate any bar plot to change titles, axis labels, and font sizes.

```
decorators = list(bar_plot =
  ggplot_decorator("bar_plot", label_text = "decorate bar plot",
    render_ui = c("title", "x_labels_cont",
                  "font_size_axis_title", "font_size_axis_text"),
    plot_options = list("title" = "TEST",
                        "x_labels_cont" = "timept1, timept2, timept3",
                        "font_size_axis_title" = "8",
                        "font_size_axis_text" = "8")
  )
)
```

This gives us a new UI that allows us to control the title and the x-axis labels and font sizes for ggplot2 modules.

Figure 2.3

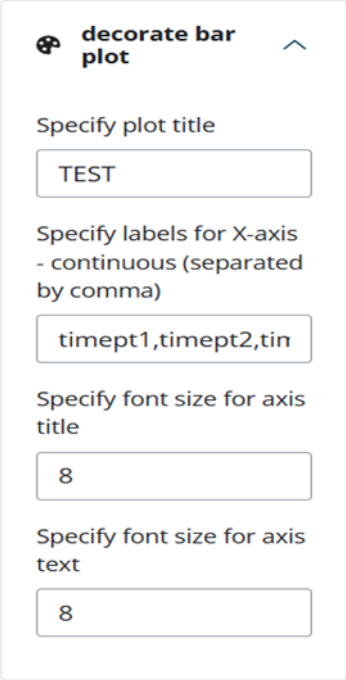
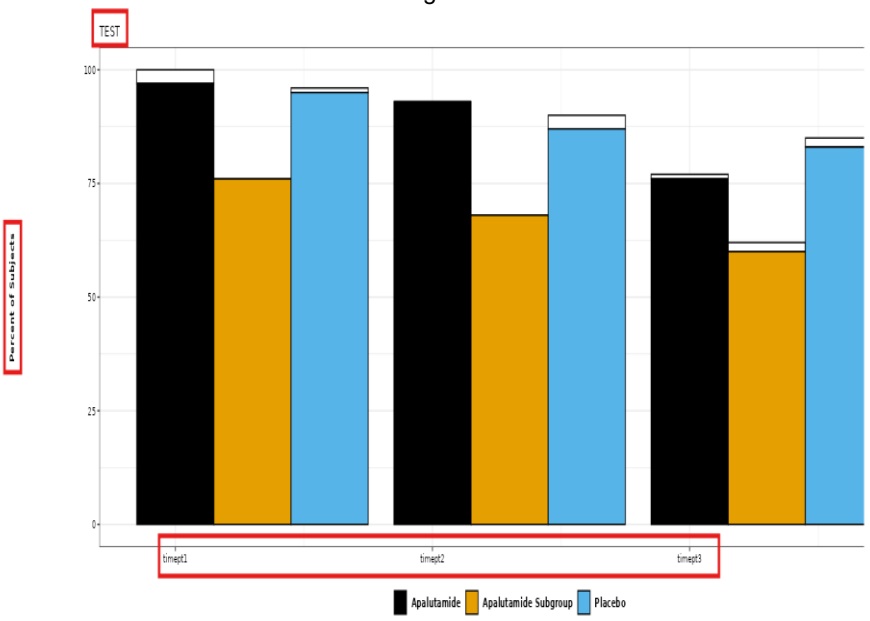


Figure 2.4



In the above plot we can see that the axis labels and titles are not desired, we may wish to change these and also make them more legible.

Figure 2.5

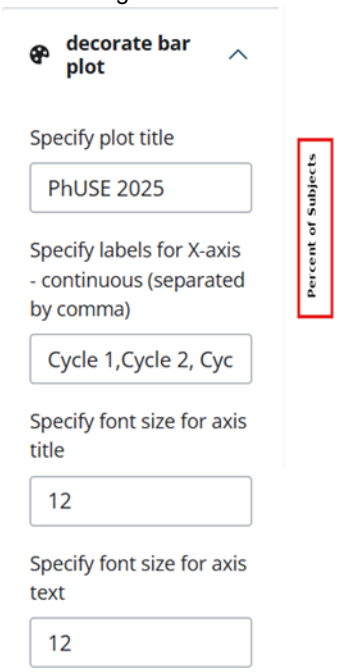
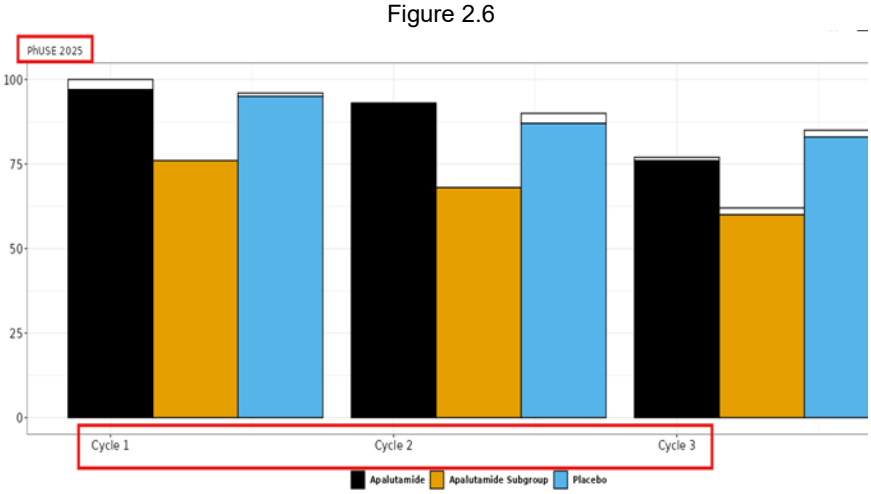


Figure 2.6



This decorator also has a multitude of other formatting options, including axis scales, to customize the plot to further meet requirements.

ENTERPRISE ACCESS CONTROL FRAMEWORK: {rAccess} INTEGRATION

The {rAccess} package provides a comprehensive access control framework, addressing critical security and compliance requirements in pharmaceutical environments. This package enables granular permission management across user groups and application components, ensuring adherence to data governance policies and regulatory standards.

We can use {rAccess} as a separate module within our application to control which users have access to which panels within the application. We can assign users as administrators for this task of controlling access. This is particularly useful when sharing these applications with functions outside of data science (e.g., clinical stakeholders). Perhaps some visualizations we have created within our programming team are more sensitive to wider groups, or we don't wish for these teams to build conclusions from any of these visualizations.

The {rAccess} implementation is fairly straightforward to set up and requires a configuration .yml file to be created. Each of the modules should be included in this file. This is also compatible with an AWS S3 bucket so the user lists for each application can be stored and managed from one central location for all deployed applications.

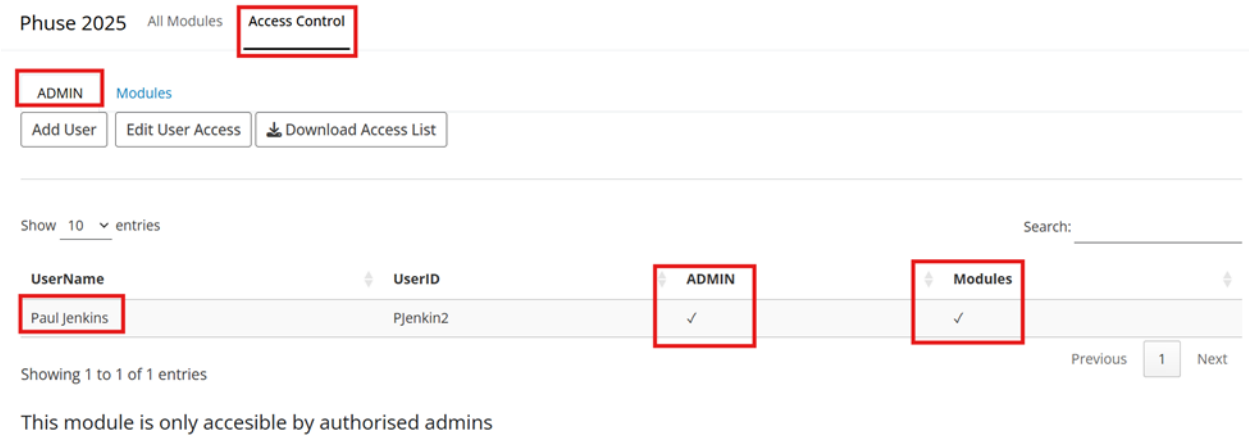
An example of the configuration .yml file (using a local environment for demo purposes) can be seen below:

Figure 3.1

```
24
25  panel_str:
26  - access_panel: ADMIN
27  - access_panel: Modules
28    access_units:
29    - unit: General information about the app
30    - unit: Data Table
31    - unit: Variable Browser
32    - unit: Disposition
33    - unit: Disposition (JJCS)
34    - unit: Analysis Sets
35    - unit: Demographic and Baseline Disease Characteristics
36    - unit: Site/Country Summary
37    - unit: Prior/Concomitant Medications
38    - unit: Overall AE Summary
39    - unit: All AE Tables by Term
40    - unit: AE Severity by Term
41    - unit: Adverse Events by Grade
42    - unit: Deaths
```

In the example below, within our 'Access Control' module a user has been added as 'ADMIN' and subsequently been given the rights to control the access for the 'Modules' panel. This is where access can be granted for everyone or individuals for all teal modules within the application.

Figure 3.2

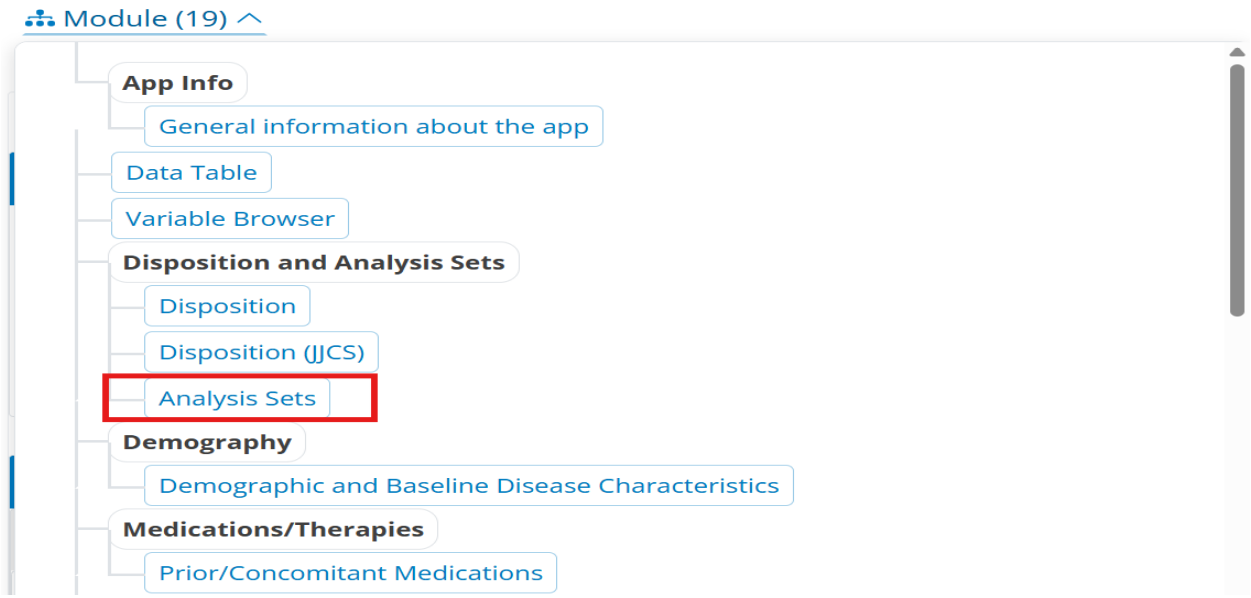


This 'Modules' tab covers all individual lowest-level child modules being used within the application. The lowest level module label used in each teal module is displayed within the access page. We can then use this page to assign 'Everyone' or individual users to have access to individual modules.

Figure 3.3

We can see this is true when loading the application as 'Paul Jenkins' - only the modules that have been granted access are visible. As expected, the 'Site/Country Summary' module is not visible, yet the 'Analysis Sets' module is accessible, which would not be accessible to everyone.

Figure 3.4



CUSTOM MODULE DEVELOPMENT ADDRESSING INDUSTRY-SPECIFIC REQUIREMENTS

Pharmaceutical organizations typically maintain proprietary analytical standards spanning the complete data lifecycle from SDTM through TLF generation. While the standard {teal.modules.clinical} package provides comprehensive analytical capabilities, organizational-specific requirements often necessitate customized implementations to ensure alignment with internal standards and regulatory expectations.

Johnson & Johnson has undertaken a systematic approach to developing custom {teal} modules that address these specialized requirements while maintaining compatibility with the broader framework ecosystem. Our development strategy emphasizes creating reusable, open-source components that can benefit the wider pharmaceutical community while meeting our specific analytical needs.

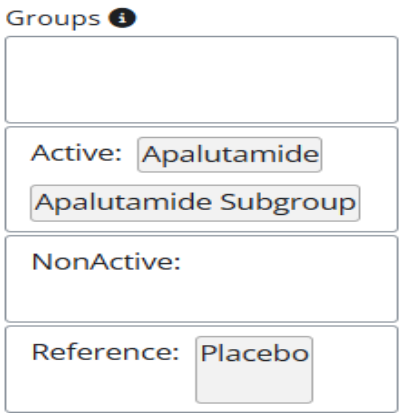
A few examples of modules created in-house are shown below.

EVENT MODULE (tm_t_events_j)

Our in-house version of the {teal} module tm_t_events (used commonly for adverse events and prior/concomitant medication tables) has more functionality and supports the requirements of our internal TLF standards. A few of the key enhancements made to this module are as follows:

- A new UI to support active, non-active, and reference treatment group selections (used to apply spanning column headers and for calculating additional required statistics)

Figure 4.1



- New UI option to add a combined active treatment column based on selection from the previous bullet point (using the assignments defined in the previous bullet point)

Figure 4.2

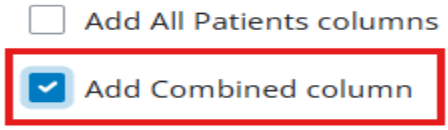


Figure 4.3

	Active Study Agent			
Body System or Organ Class	Apalutamide	Apalutamide Subgroup	Combined	Placebo
Preferred Term, n (%)	(N=72)	(N=96)	(N=168)	(N=86)
Subjects with >= 1 adverse event	36 (50.0%)	47 (49.0%)	83 (49.4%)	32 (37.2%)
Cardiac disorders	3 (4.2%)	5 (5.2%)	8 (4.8%)	5 (5.8%)
ATRIAL FIBRILLATION	0	1 (1.0%)	1 (0.6%)	0

- New UI allowing the inclusion of risk difference calculations in additional table settings (using the reference treatment defined in the first bullet point)

Figure 4.4

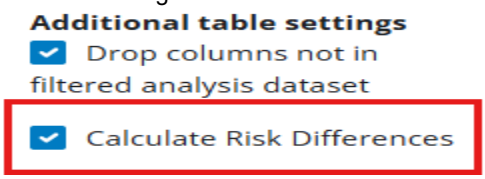
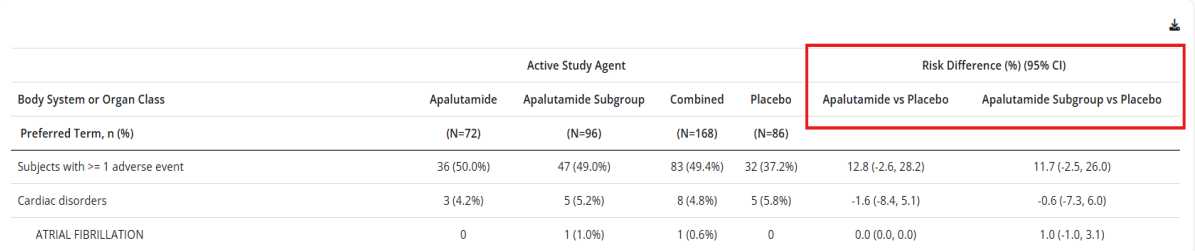


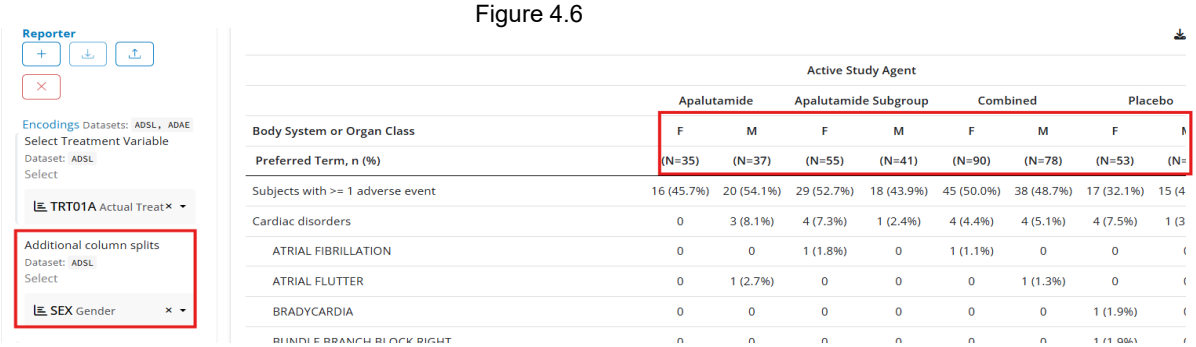
Figure 4.5



	Active Study Agent				Risk Difference (%) (95% CI)	
Body System or Organ Class	Apalutamide	Apalutamide Subgroup	Combined	Placebo	Apalutamide vs Placebo	Apalutamide Subgroup vs Placebo
Preferred Term, n (%)	(N=72)	(N=96)	(N=168)	(N=86)		
Subjects with >= 1 adverse event	36 (50.0%)	47 (49.0%)	83 (49.4%)	32 (37.2%)	12.8 (-2.6, 28.2)	11.7 (-2.5, 26.0)
Cardiac disorders	3 (4.2%)	5 (5.2%)	8 (4.8%)	5 (5.8%)	-1.6 (-8.4, 5.1)	-0.6 (-7.3, 6.0)
ATRIAL FIBRILLATION	0	1 (1.0%)	1 (0.6%)	0	0.0 (0.0, 0.0)	1.0 (-1.0, 3.1)

- New UI to accept 1 additional column split:

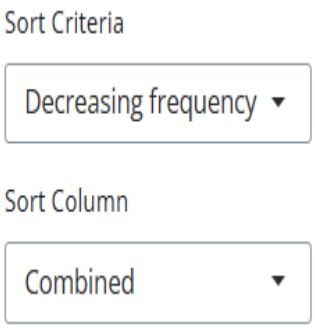
Figure 4.6



	Active Study Agent							
	Apalutamide		Apalutamide Subgroup		Combined		Placebo	
Body System or Organ Class	F	M	F	M	F	M	F	M
Preferred Term, n (%)	(N=35)	(N=37)	(N=55)	(N=41)	(N=90)	(N=78)	(N=53)	(N=
Subjects with >= 1 adverse event	16 (45.7%)	20 (54.1%)	29 (52.7%)	18 (43.9%)	45 (50.0%)	38 (48.7%)	17 (32.1%)	15 (4
Cardiac disorders	0	3 (8.1%)	4 (7.3%)	1 (2.4%)	4 (4.4%)	4 (5.1%)	4 (7.5%)	1 (3
ATRIAL FIBRILLATION	0	0	1 (1.8%)	0	1 (1.1%)	0	0	(
ATRIAL FLUTTER	0	1 (2.7%)	0	0	0	1 (1.3%)	0	(
BRADYCARDIA	0	0	0	0	0	0	1 (1.9%)	(

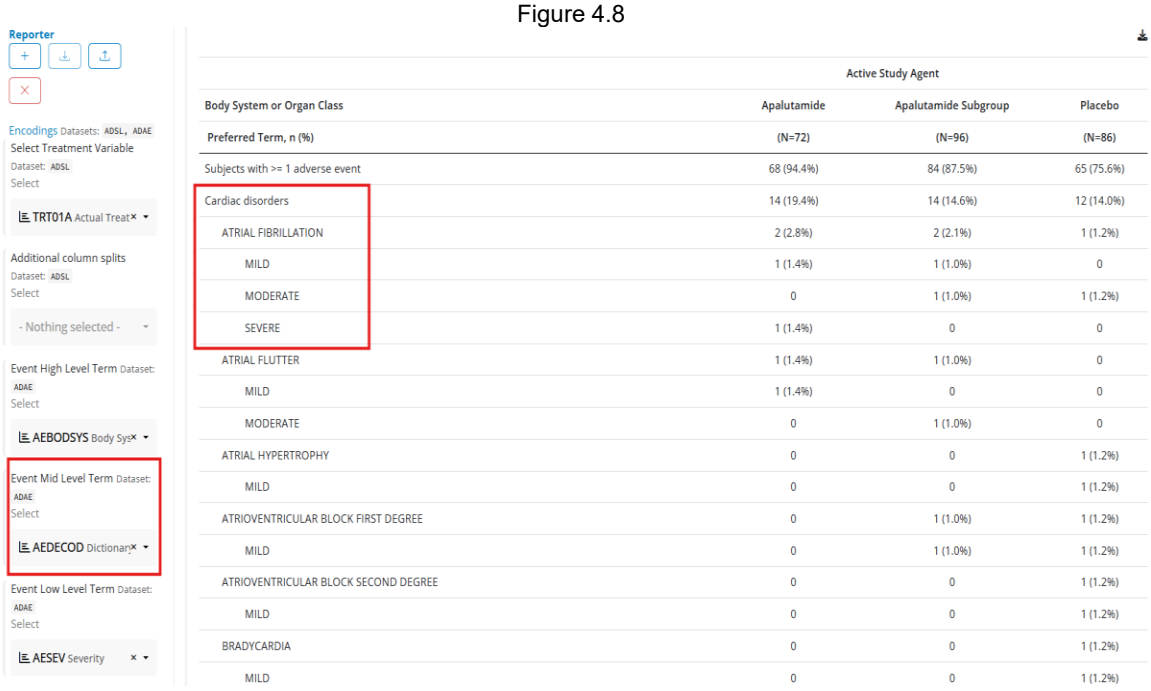
- A new UI that allows for default sort criteria (based on factor levels), alphabetical sorting, or sorting based on descending frequency of any treatment arm the user requests. An example is shown below.

Figure 4.7



- The introduction of an MLT argument to support 3 tiers of levels (often used in CM tables and also in AE tables by SOC/PT/toxicity)

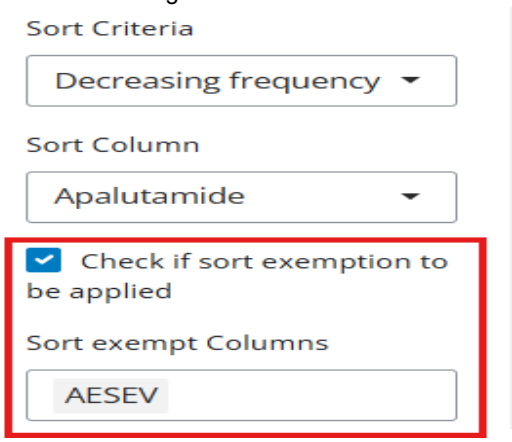
Figure 4.8



	Active Study Agent		
Body System or Organ Class	Apalutamide	Apalutamide Subgroup	Placebo
Preferred Term, n (%)	(N=72)	(N=96)	(N=86)
Subjects with >= 1 adverse event	68 (94.4%)	84 (87.5%)	65 (75.6%)
Cardiac disorders	14 (19.4%)	14 (14.6%)	12 (14.0%)
ATRIAL FIBRILLATION	2 (2.8%)	2 (2.1%)	1 (1.2%)
MILD	1 (1.4%)	1 (1.0%)	0
MODERATE	0	1 (1.0%)	1 (1.2%)
SEVERE	1 (1.4%)	0	0
ATRIAL FLUTTER	1 (1.4%)	1 (1.0%)	0
MILD	1 (1.4%)	0	0
MODERATE	0	1 (1.0%)	0
ATRIAL HYPERTROPHY	0	0	1 (1.2%)
MILD	0	0	1 (1.2%)
ATRIOVENTRICULAR BLOCK FIRST DEGREE	0	1 (1.0%)	1 (1.2%)
MILD	0	1 (1.0%)	1 (1.2%)
ATRIOVENTRICULAR BLOCK SECOND DEGREE	0	0	1 (1.2%)
MILD	0	0	1 (1.2%)
BRADYCARDIA	0	0	1 (1.2%)
MILD	0	0	1 (1.2%)

- Ability to ignore the sort criteria on any of the above 3 tiers (llt, mlt, or hlt), which is particularly useful for the AE severity table above where you want to retain default ordering of MILD, MODERATE, SEVERE and not rearrange all grades/severities by descending frequencies

Figure 4.9



NEW LABORATRY TABLE FORMATS

As an extension of existing laboratory table layouts available in {teal}, we have created some of our own teal modules to facilitate these requirements.

SUMMARY TABLE MODULE (tm_t_summary_by_j)

We can see below that the tm_t_summary_by_j module we have created allows for N, Mean, and Mean Change from baseline to be displayed as columns, while presenting the laboratory parameters and analysis visits in the row space.

Figure 5.1

	Apalutamide			Apalutamide Subgroup			Placebo		
Parameter	N=72			N=96			N=86		
Analysis Visit	n/N (%)	Mean (95% CI)	Mean Change from Baseline(95% CI)	n/N (%)	Mean (95% CI)	Mean Change from Baseline(95% CI)	n/N (%)	Mean (95% CI)	Mean Change from Baseline(95% CI)
Alanine Aminotransferase (U/L)									
Baseline	72/72 (100.0%)	19.2 (16.8, 21.6)		94/94 (100.0%)	18.1 (16.4, 19.9)		86/86 (100.0%)	17.6 (15.6, 19.5)	
Cycle 02	70/70 (100.0%)	21.2 (19.1, 23.3)	2.2 (0.6, 3.7)	88/88 (100.0%)	20.7 (18.5, 22.9)	2.3 (0.8, 3.8)	83/83 (100.0%)	18.0 (15.3, 20.7)	0.3 (-1.4, 2.0)
Cycle 03	72/72 (100.0%)	21.3 (19.1, 23.5)	2.3 (0.6, 3.9)	72/72 (100.0%)	17.5 (15.7, 19.3)	-0.6 (-1.6, 0.4)	79/79 (100.0%)	18.7 (15.8, 21.5)	0.8 (-1.1, 2.7)
Cycle 04	66/66 (100.0%)	21.2 (18.9, 23.6)	1.7 (0.2, 3.2)	62/62 (100.0%)	17.0 (15.0, 19.0)	-1.1 (-2.4, 0.2)	73/73 (100.0%)	17.0 (14.6, 19.3)	-0.4 (-2.1, 1.3)
Cycle 05	56/56 (100.0%)	22.8 (18.1, 27.5)	3.3 (-1.3, 7.9)	60/60 (100.0%)	17.6 (15.6, 19.7)	0.0 (-1.5, 1.5)	72/72 (100.0%)	16.7 (14.5, 18.9)	-1.0 (-2.2, 0.3)
Cycle 06	-	NE (NE, NE)	NE (NE, NE)	1/1 (100.0%)	16.0 (NE, NE)	-2.0 (NE, NE)	1/1 (100.0%)	73.0 (NE, NE)	23.0 (NE, NE)
Cycle 07	50/50 (100.0%)	21.0 (18.1, 23.9)	0.9 (-1.6, 3.4)	51/51 (100.0%)	18.5 (14.9, 22.0)	0.4 (-2.4, 3.2)	67/67 (100.0%)	18.0 (15.8, 20.2)	0.2 (-1.6, 2.0)
Cycle 08	-	NE (NE, NE)	NE (NE, NE)	-	NE (NE, NE)	NE (NE, NE)	1/1 (100.0%)	17.0 (NE, NE)	-5.0 (NE, NE)
Cycle 09	37/37 (100.0%)	19.6 (17.0, 22.1)	0.0 (-2.4, 2.4)	42/42 (100.0%)	17.3 (15.0, 19.6)	0.7 (-1.1, 2.5)	68/68 (100.0%)	17.1 (15.3, 18.8)	-0.6 (-2.3, 1.0)
Cycle 10	1/1 (100.0%)	12.0 (NE, NE)	1.0 (NE, NE)	2/2 (100.0%)	17.0 (-8.4, 42.4)	-2.0 (-27.4, 23.4)	-	NE (NE, NE)	NE (NE, NE)
Cycle 11	31/31 (100.0%)	19.6 (17.1, 22.1)	-0.3 (-3.3, 2.8)	30/30 (100.0%)	16.7 (14.3, 19.1)	1.1 (-0.8, 3.0)	65/65 (100.0%)	16.1 (14.4, 17.7)	-1.8 (-3.3, -0.2)
Cycle 12	1/1 (100.0%)	42.0 (NE, NE)	32.0 (NE, NE)	-	NE (NE, NE)	NE (NE, NE)	1/1 (100.0%)	13.0 (NE, NE)	1.0 (NE, NE)
Cycle 13	30/30 (100.0%)	21.0 (17.7, 24.2)	0.3 (-2.7, 3.3)	26/26 (100.0%)	18.2 (14.5, 21.9)	1.8 (-0.5, 4.2)	57/57 (100.0%)	17.9 (13.7, 22.0)	-0.1 (-4.4, 4.2)
Cycle 15	-	NE (NE, NE)	NE (NE, NE)	-	NE (NE, NE)	NE (NE, NE)	1/1 (100.0%)	9.0 (NE, NE)	0.0 (NE, NE)
Cycle 17	-	NE (NE, NE)	NE (NE, NE)	1/1 (100.0%)	15.0 (NE, NE)	1.0 (NE, NE)	1/1 (100.0%)	12.0 (NE, NE)	-3.0 (NE, NE)
Cycle 18	-	NE (NE, NE)	NE (NE, NE)	1/1 (100.0%)	20.0 (NE, NE)	0.0 (NE, NE)	-	NE (NE, NE)	NE (NE, NE)
End Of Treatment	27/27 (100.0%)	18.9 (16.1, 21.6)	-1.9 (-4.6, 0.7)	25/25 (100.0%)	17.8 (13.9, 21.8)	1.6 (-1.0, 4.3)	57/57 (100.0%)	16.0 (14.4, 17.6)	-1.8 (-3.5, -0.1)

SHIFT TABLE MODULE (tm_t_shift_grade_j)

Similarly, we can see below that the tm_t_shift_grade_j module we have created allows a different shift table layout from what is offered via the {teal} module TM_T_SHIFT_BY_GRADE. Again, this allows our teams to create specific layouts that are aligned with our internal standards within their {teal} applications.

Figure 5.2

Laboratory Test		Baseline			
Study Visit	N	Low	Normal	High	Total
Apalutamide	72				
Apalutamide Subgroup	96				
Placebo	86				
Alanine Aminotransferase (U/L)					
Baseline					
Apalutamide	72				
Low		0	0	0	0
Normal		0	68	0	68
High		0	0	4	4
Total		0	68	4	72
Apalutamide Subgroup	94				
Low		1	0	0	1
Normal		0	89	1	90
High		0	1	2	3
Total		1	90	3	94
Placebo	86				
Low		0	0	0	0
Normal		0	82	0	82
High		0	0	4	4

CONCLUSION

In this paper we have explored recent developments in {teal} framework including improved user interface components and navigation enhancements, examined sophisticated architectural improvements through transformators and decorators, demonstrated access control capabilities via {rAccess} integration, and showcased custom module development that addresses industry-specific analytical requirements.

The transformators and decorators represent a paradigm shift in framework extensibility, enabling organizations to implement complex customizations while maintaining system integrity. Johnson & Johnson's implementation experience illustrates how these enhancements facilitate the development of specialized analytical capabilities, while preserving compatibility with the broader {teal} ecosystem.

Most importantly, this work exemplifies the strategic value of cross-industry collaboration in driving pharmaceutical innovation. The `Teal Enhancements for Cross-Industry Adoption` has been instrumental in creating a collaborative environment where pharmaceutical companies can share challenges, contribute solutions, and collectively advance the state of clinical data analysis.

Through initiatives like the {uteals} package and standardized module development approaches, our cross-industry collaboration ensures that enhancements benefit the entire pharmaceutical community rather than remaining siloed within individual organizations. This collaborative approach not only accelerates development timelines but also promotes standardization, improves quality through peer review, and reduces duplicated effort across the industry.

We encourage continued participation in the `Teal Enhancements for Cross-Industry Adoption` and broader {teal} community to further advance these collaborative efforts. The future of pharmaceutical data visualization lies in our collective ability to share knowledge, contribute innovations, and build upon each other's work to ultimately accelerate the delivery of life-saving medicines to patients worldwide.

REFERENCES

Insights Engineering. (2025). *{teal} News and Changelog*. GitHub Pages. https://insightsengineering.github.io/teal/latest-tag/news/index.html

PHUSE Working Group. (2025). *Teal Enhancements for Cross-Industry Adoption*. PHUSE Wiki. https://advance.hub.phuse.global/wiki/spaces/WEL/pages/30441473/Teal+Enhancements+for+Cross-Industry+Adoption

Roche. (2025). *PAP_OS13: {teal} Framework Developments*. PHUSE US Connect 2025, Orlando. https://phuse.s3.eu-central-1.amazonaws.com/Archive/2025/Connect/US/Orlando/PAP_OS13.pdf

PHUSE Working Group. (2025). *{uteals}: Utility Functions for Teal Applications*. GitHub Repository. https://github.com/phuse-org/uteals

Johnson & Johnson. (2025). *{rAccess}: Access Control Framework for R Applications*. GitHub Pages. https://johnsonandjohnson.github.io/rAccess/

ACKNOWLEDGMENTS

The authors would like to acknowledge Nina Qi and Dony Unard or their presentation on {teal} framework developments at PHUSE US Connect 2025. We have endeavored to maintain consistency with all {teal} community communication, highlighting collective commitment to advancing pharmaceutical data visualization through coordinated collaboration.

Special thanks to the PHUSE Working Group members and the broader {teal} community for their ongoing collaboration and feedback that continues to drive innovation in pharmaceutical data visualization through unified messaging and shared objectives.

We also acknowledge the use of AI assistance in manuscript preparation, including structural organization, proofreading, and formatting.

RECOMMENDED READING

- PHUSE US Connect 2025 Presentation: OS13 - Unleashing the Power of {teal}: New Features and Cross-Industry Collaboration.
- teal video: A Complete Guide to Getting Started with teal
- teal web manual: https://insightsengineering.github.io/teal
- teal Github repository: https://github.com/insightsengineering/teal
- teal.gallery demo apps: https://insightsengineering.github.io/teal.gallery/
- NEST R product family website: https://insightsengineering.github.io/nest/

CONTACT INFORMATION

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

Author Name: Peyman Eshghi
Company: Johnson and Johnson
Email: peshghi@its.jnj.com

Co-Author Name: Paul Jenkins
Company: Johnson and Johnson
Email: :pjenkin2@its.jnj.com