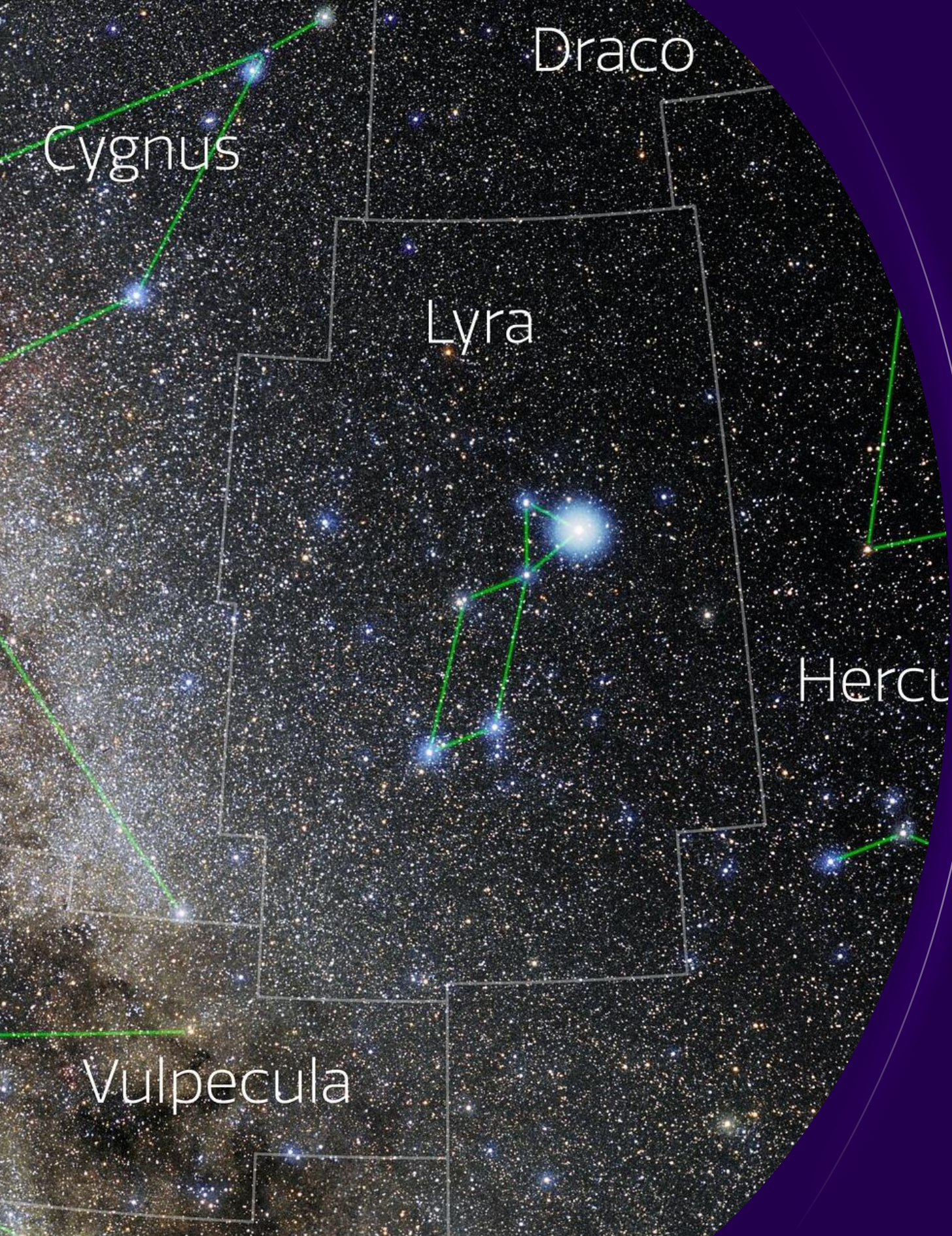


The Sanofi logo is centered on a dark blue background. It consists of the word "sanofi" in a white, lowercase, sans-serif font. Behind the text are three concentric circles of the same dark blue color as the background, creating a subtle circular pattern.

sanofi



Introduction to {lyra}

Huan Lu
Zhenhuan Yang

May 15th, 2026

The views and opinions expressed in this presentation are those of the author(s) and do not necessarily reflect the official policy or position of *PhUSE SDE 2026 – APAC, Beijing, China*

sanofi

Generated by Concierge AI

Background

`{rtables}`

- Built for complex, human-readable reporting tables
- Designed for clinical and regulatory reporting
- Separates layout definition from table generation
- Supports flexible, symbolic table structures
- Preserves semantic and computational context
- Facilitates formatting, pagination, and QC

`{gtsummary}`

- More elegant and streamlined for publication-ready tables
- Uses sensible defaults, reducing programming effort for standard outputs
- Faster to build, review, and maintain with concise syntax
- Easier to use for common summary and analytic tables
- Better supports reproducible reporting, including inline reporting in R Markdown
- Strong native support for summary tables and regression tables
- Integrates well with modern R workflows through `gt`, `broom`, and `labelled`
- Offers flexible customization without requiring the same level of low-level table construction

Comparison

`{rtables}`-based

≈ 2,000,000 bytes

`{gtsummary}`-based

≈ 300,000 bytes

based on a pack of CSR standard output template programs

Inefficiency

Layout

Complex reporting layouts, including hierarchical structures, custom row organization, and display control needed for clinical templates from {gtsummary} are different

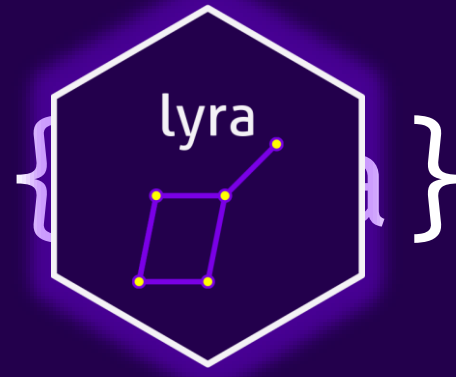
Format

Standardizes formatting behaviors such as statistic display, spacing, indentation, and no-data handling in outputs need to align with internal reporting standards

Output

Production-oriented reporting workflow by improving export readiness, template consistency, and integration with internal output delivery to downstream system must be addressed

Introducing



Overview

Inspired by the constellation Lyra (*/ˈlaɪrə/*), which symbolizes refinement and the pursuit of perfection, this package builds upon open-source foundations and provides supplementary functions to `{gtsummary}`, specifically for trial reporting at Sanofi. It adds elegant enhancements and optimized structures, bringing a touch of brilliance to your clinical trial reporting workflow.



Wrapper

Encapsulates routine reporting steps into reusable functions for greater efficiency and consistency



Helper

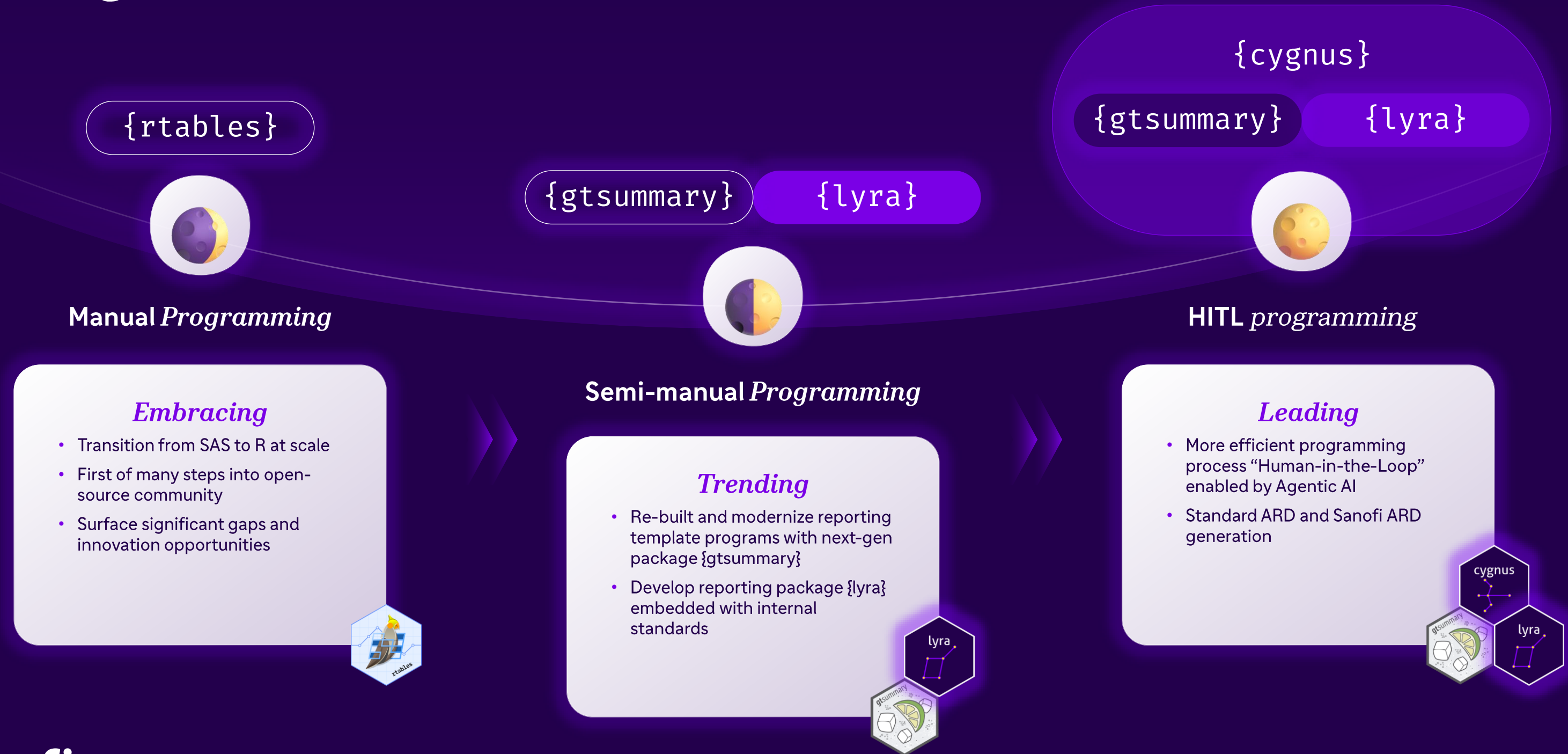
Supports standardized output formatting while allowing flexible customization for different reporting needs



Organizer

Consolidates shared functions into one package to improve reuse, maintenance, and development scalability

Design

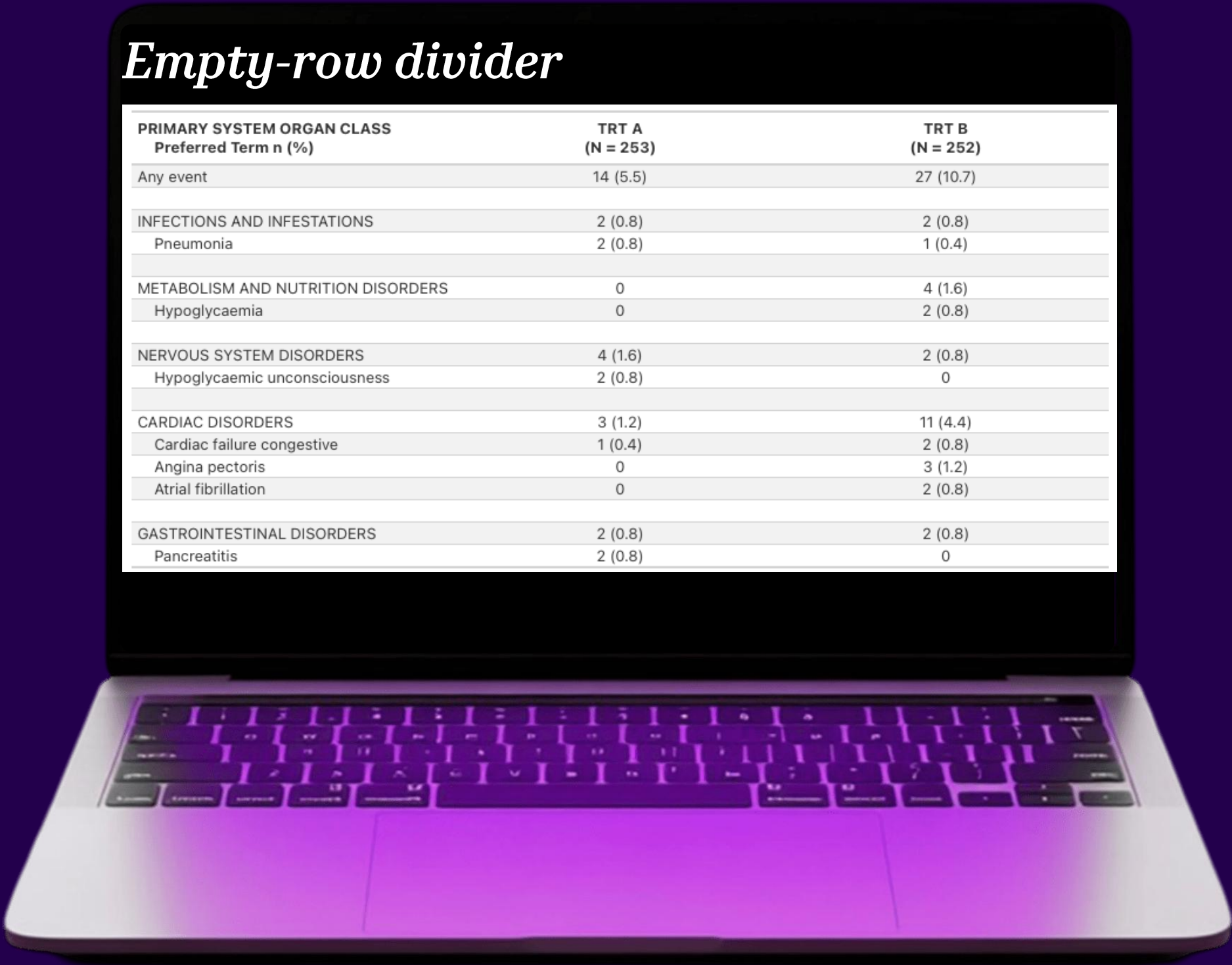


Example

```
tbl <- tbl_hierarchical(  
  data = anl_adae,  
  denominator = anl_adsl,  
  id = USUBJID,  
  by = trt01a,  
  variables = c(aebodsys, aedecod),  
  statistic = ~ "{n} ({p})",  
  digits = everything() ~ list(p = 1),  
  overall_row = TRUE,  
  label = list(..ard_hierarchical_overall.. = "Any event",  
               aebodsys ~ "PRIMARY SYSTEM ORGAN CLASS",  
               aedecod ~ "Preferred Term n (%)")  
) |>  
  filter_hierarchical(!rlang::parse_expr(cutoff)) |>  
  lyra::modify_stat_fmt() |>  
  lyra::arrange_hierarchy(by = "stat_1", descending = TRUE) |>  
  modify_header(all_stat_cols() ~ "**{level} \n(N = {n})**") |>  
  lyra::add_empty_row(rows = variable %in% "aebodsys",  
                      position = "before") |>  
  remove_footnote_header(columns = everything())
```

Empty-row divider

PRIMARY SYSTEM ORGAN CLASS Preferred Term n (%)	TRT A (N = 253)	TRT B (N = 252)
Any event	14 (5.5)	27 (10.7)
INFECTIONS AND INFESTATIONS	2 (0.8)	2 (0.8)
Pneumonia	2 (0.8)	1 (0.4)
METABOLISM AND NUTRITION DISORDERS	0	4 (1.6)
Hypoglycaemia	0	2 (0.8)
NERVOUS SYSTEM DISORDERS	4 (1.6)	2 (0.8)
Hypoglycaemic unconsciousness	2 (0.8)	0
CARDIAC DISORDERS	3 (1.2)	11 (4.4)
Cardiac failure congestive	1 (0.4)	2 (0.8)
Angina pectoris	0	3 (1.2)
Atrial fibrillation	0	2 (0.8)
GASTROINTESTINAL DISORDERS	2 (0.8)	2 (0.8)
Pancreatitis	2 (0.8)	0



Example

```
tbl <- tbl_hierarchical(  
  data = anl_adae,  
  denominator = anl_adsl,  
  id = USUBJID,  
  by = trt01a,  
  variables = c(aebodsys, aehlgt, aehlt, aedecod),  
  statistic = ~ "{n} ({p})",  
  digits = everything() ~ list(p = 1),  
  overall_row = TRUE,  
  label = list(..ard_hierarchical_overall.. = "Any event",  
               aebodsys ~ "PRIMARY SYSTEM ORGAN CLASS",  
               aehlgt ~ "HLGT: High Level Group Term",  
               aehlt ~ "HLT: High Level Term",  
               aedecod ~ "Preferred Term n (%)")  
) |>  
  filter_hierarchical(!rlang::parse_expr(cutoff)) |>  
  lyra::modify_stat_fmt() |>  
  sort_hierarchical("alphanumeric") |>  
  lyra::arrange_hierarchy(by = "overall", descending = TRUE) |>  
  modify_header(all_stat_cols() ~ "**{level} \n(N = {n})**") |>  
  lyra::add_empty_row(rows = variable %in% "aebodsys",  
                      position = "before") |>  
  remove_footnote_header(columns = everything())
```

Hierarchical Sorting

PRIMARY SYSTEM ORGAN CLASS HLGT: High Level Group Term HLT: High Level Term Preferred Term n (%)	TRT A (N = 253)	TRT B (N = 252)
Any event	14 (5.5)	27 (10.7)
CARDIAC DISORDERS	3 (1.2)	11 (4.4)
HLGT: Coronary artery disorders	0	4 (1.6)
HLT: Ischaemic coronary artery disorders	0	3 (1.2)
Angina pectoris	0	3 (1.2)

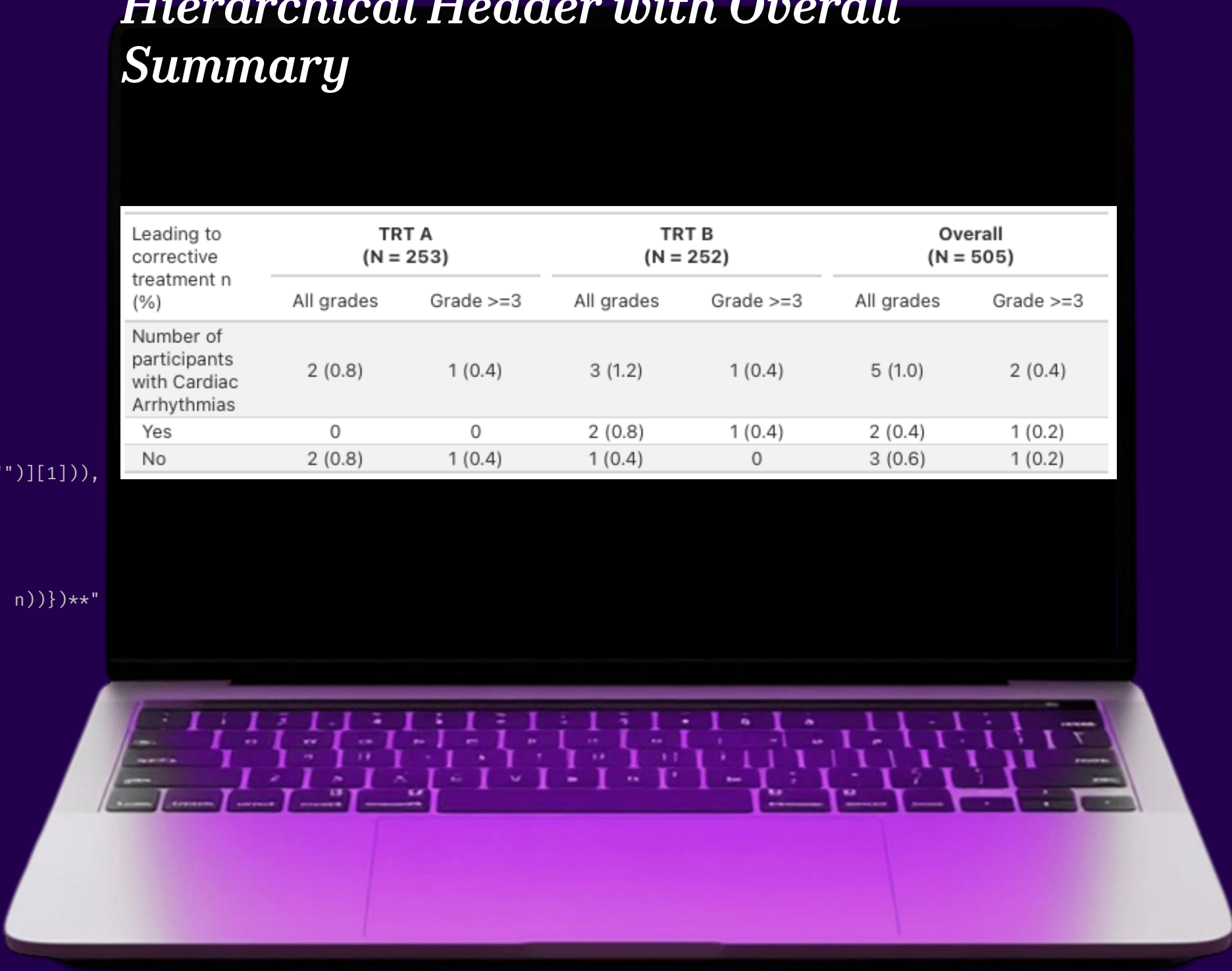


Example

```
tbl <- tbl_strata(  
  data = anl_data_all,  
  strata = trt01a,  
  .tbl_fun =  
    ~ .x |>  
    tbl_summary(  
      by = toxgrn_grp,  
      include = c(aeconrt_all, aeconrt_y, aeconrt_n),  
      statistic = list(all_categorical() ~ "{n} ({p})"),  
      digits = list(all_categorical() ~ c(0, 1)),  
      label = list(aeconrt_all = paste0("Number of participants with ",  
str_to_title(adae[[sub("CD$", "NAM", smq_val)]] [which(adae[[smq_val]] != "")][1])),  
        aeconrt_y = "Yes",  
        aeconrt_n = "No")  
    ),  
  .header = "**{strata} \n(N = {with(anl_adsl |> count(trt01a), c(sum(n), n))})**"  
  
  ) |>  
  lyra::modify_stat_fmt() |>  
  .....  
)
```

Hierarchical Header with Overall Summary

Leading to corrective treatment n (%)	TRT A (N = 253)		TRT B (N = 252)		Overall (N = 505)	
	All grades	Grade >=3	All grades	Grade >=3	All grades	Grade >=3
Number of participants with Cardiac Arrhythmias	2 (0.8)	1 (0.4)	3 (1.2)	1 (0.4)	5 (1.0)	2 (0.4)
Yes	0	0	2 (0.8)	1 (0.4)	2 (0.4)	1 (0.2)
No	2 (0.8)	1 (0.4)	1 (0.4)	0	3 (0.6)	1 (0.2)



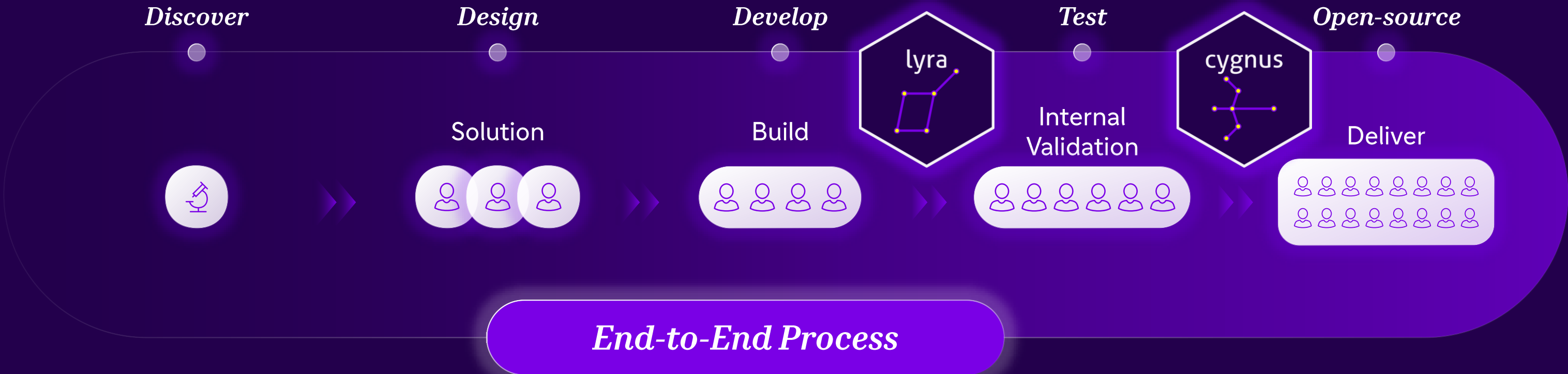
Example

```
if(nrow(anl_adae) == 0) {
  tbl <- as_gtsummary(anl_adae) |>
  lyra::modify_no_data(label = "No data")
} else {
  tbl <- tbl_hierarchical(
    data = anl_adae,
    denominator = anl_adsl,
    id = USUBJID,
    by = trt01a,
    variables = c(aebodsys, aedecod),
    statistic = ~ "{n} ({p})",
    digits = everything() ~ list(p = 1),
    overall_row = TRUE,
    label = list(..ard_hierarchical_overall.. = "Any event",
                 aebodsys ~ "PRIMARY SYSTEM ORGAN CLASS",
                 aedecod ~ "Preferred Term n (%)")
  ) |>
  filter_hierarchical(!rlang::parse_expr(cutoff)) |>
  lyra::modify_stat_fmt() |>
  lyra::arrange_hierarchy(by = "stat_1", descending = TRUE) |>
  modify_header(all_stat_cols() ~ "**{level} \n(N = {n})**") |>
  lyra::add_empty_row(rows = variable %in% "aebodsys",
                      position = "before") |>
  remove_footnote_header(columns = everything())
```

Null Report

No data

Roadmap



Maintenance

Automation

As {lyra} evolved, its release preparation became a repeated and multi-step process, requiring consistent updates across version metadata, release notes, documentation references, and publishing steps...

- DESCRIPTION
- NEWS.md
- _pkgdown.yml
- GitHub Release Notes
- ...

Package

DESCRIPTION

R/

tests/

man/

vignettes/

data/

NAMESPACE

SETUP

WRITE CODE

TEST

DOCUMENT

TEACH

ADD DATA

ORGANIZE

Why we built a release agent?

Before

*Manual
Repetitive
Easy to miss
inconsistent*

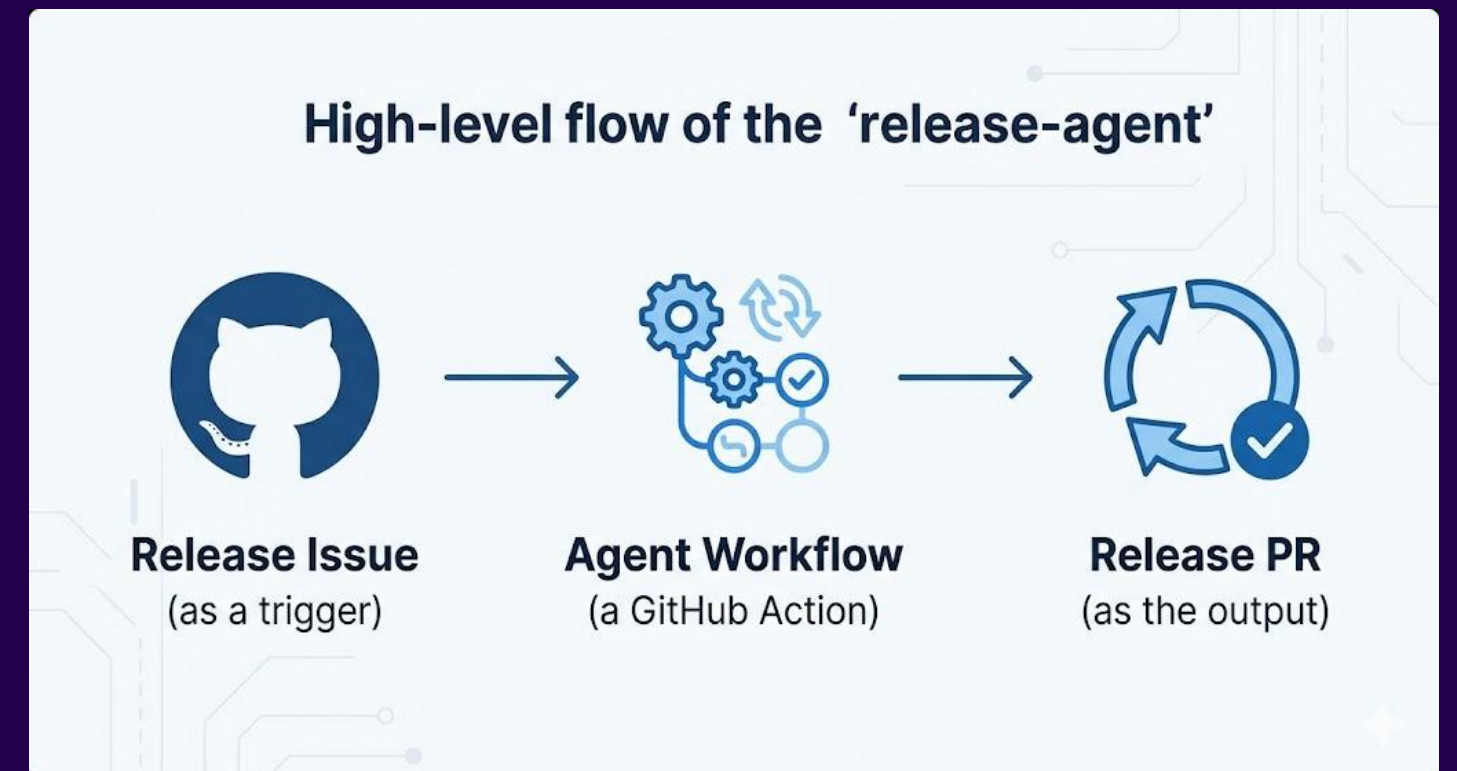
After

**Issue-driven
Standardized
Reviewable
Reusable**

Release-agent

What is it?

Release-agent is a **GitHub Action** designed to automate the heavy lifting of R package release preparation



Release-agent

release-agent



Manual & Inconsistent Preparation

Standardized workflow provides a clean, diff-based view for maintainers to approve



Scattered Updates

Synchronizes DESCRIPTION, NEWS, and docs in one go



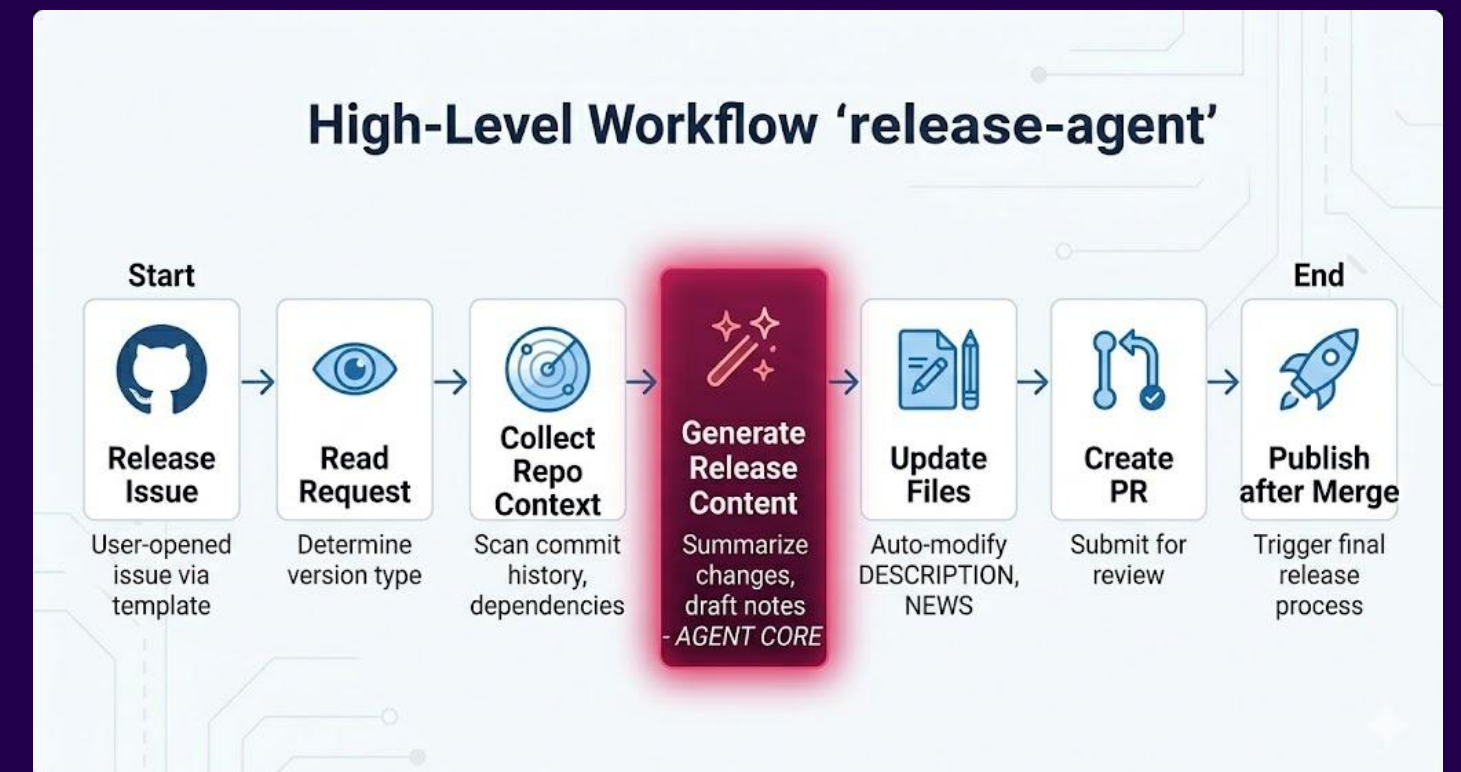
Harder Review

Shifts focus from "checking formats" to "validating content"

Release-agent

What the agent does?

It turns a release request into structured release content and a reviewable pull request



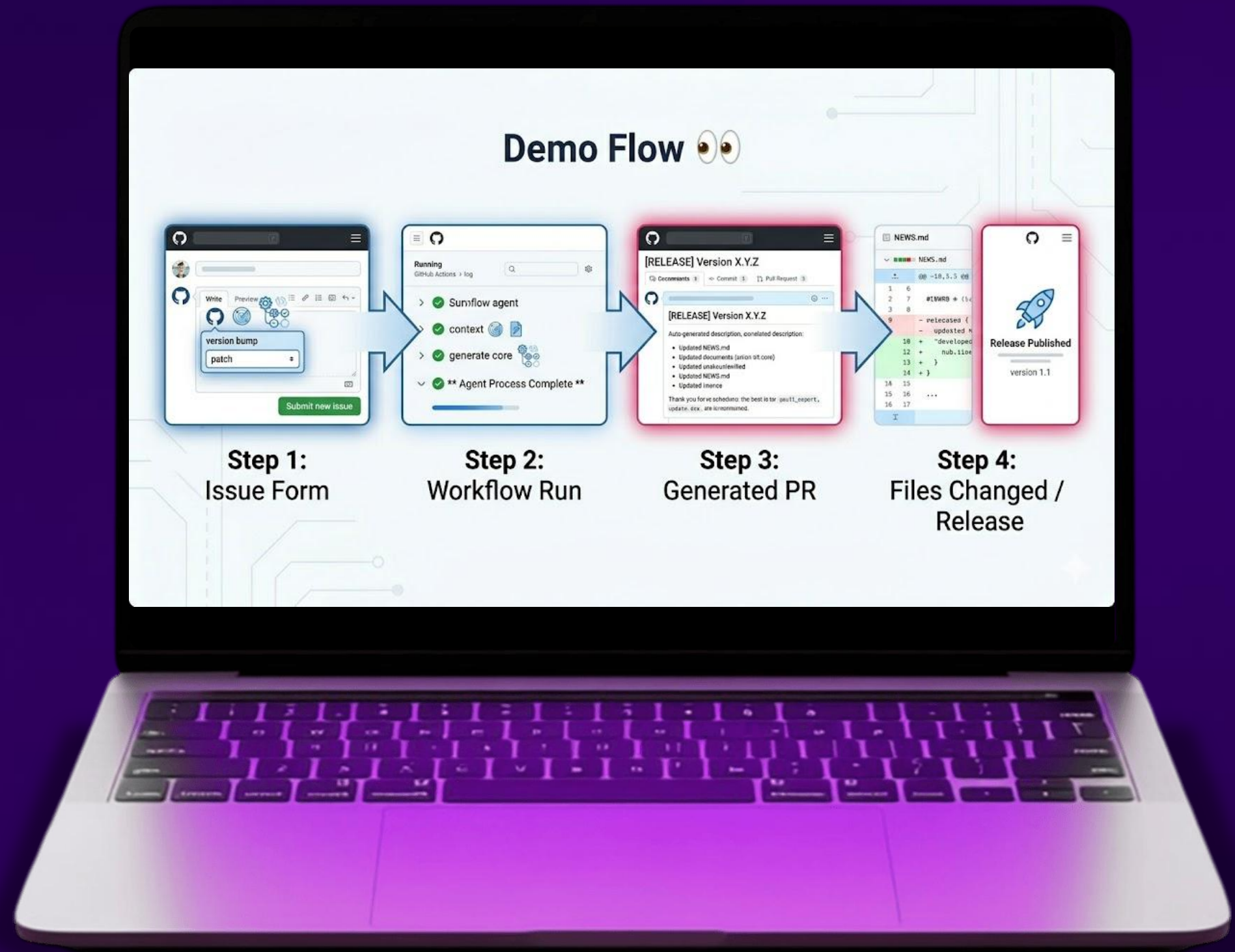
Release-agent

How {lyra} uses this action?

{lyra} uses this action to trigger an issue-driven workflow that prepares a release PR and continues the publish steps after merge



Demo



Why this matters?



For Maintainers
90%

Less Manual Overhead



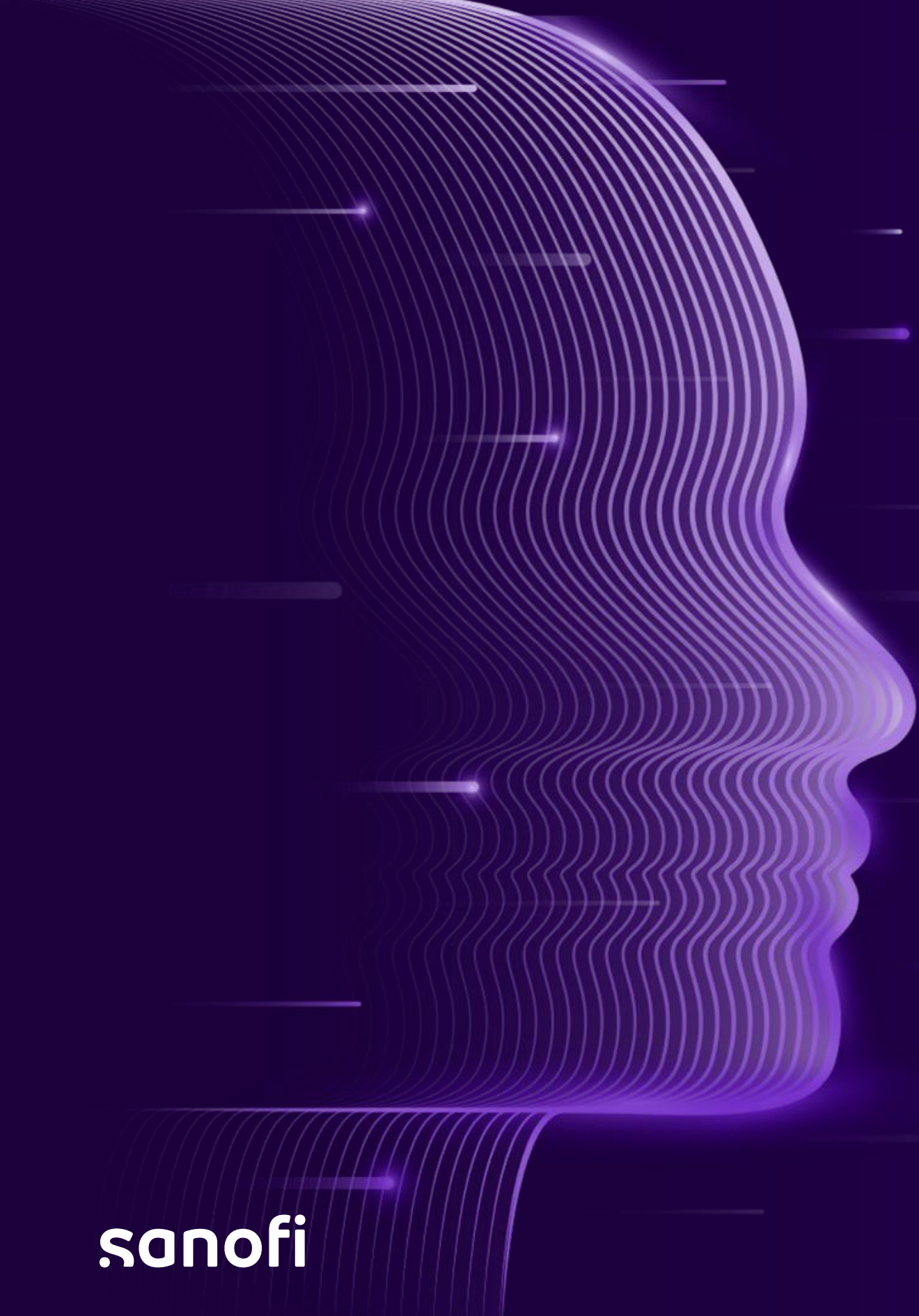
For Teams
100%

Metadata Consistency



For Reviewers
3x

Faster Review Cycles



Faster
More Consistent
Easier to Review

*The goal is not blind automation, but
better release preparation*



Q & A

sanofi

Generated by Concierge AI

Thank You

sanofi

Generated by Concierge AI