ARS for Test-Driven Development

Presented by Yann Féat, Statistician, mainanalytics GmbH

Meet the Speaker

Yann Féat

Title: Statistician

Organization: mainanalytics GmbH

Yann Féat is a dedicated statistician with over five years of experience in statistics, software engineering, and machine learning, including more than two years in clinical research at mainanalytics GmbH. Yann's passion for statistical reporting and automation drives him to improve the efficiency and accuracy of clinical trial analyses. He holds a Master's degree in Software Engineering and Applied Mathematics and actively contributes to the Regulatory R Package Repository, R Validation Hub.

Disclaimer and Disclosures

- 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 CDISC.
- The author(s) have no real or apparent conflicts of interest to report.

Introduction

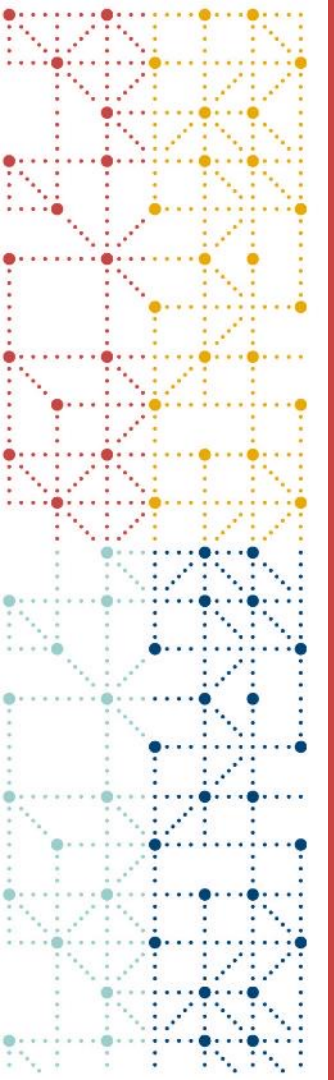
Test-driven development (TDD)

development methodology where tests are written before the main code to ensure that all requirements are met

Analysis results standard (ARS)

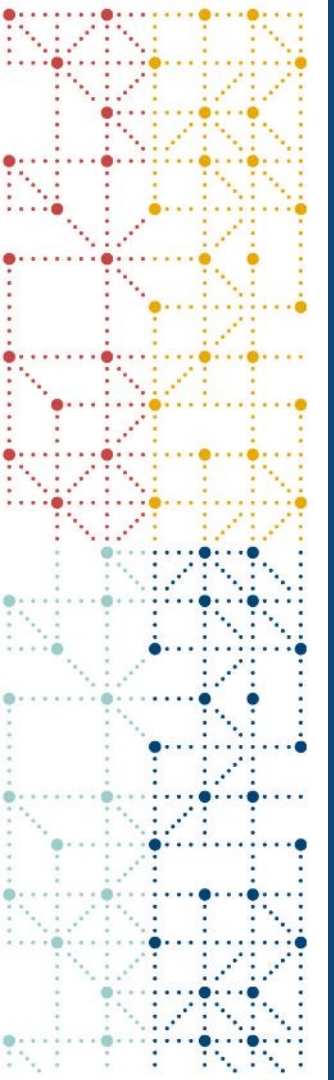
data model to combine results of statistical analyses with their metadata in a machine-readable way

Combining TDD and ARS into a comprehensive workflow to ensure better consistency, accuracy and efficiency in static reports and in interactive apps



Agenda

1. Context and Background
2. Introduction to ARS for TDD and their Benefits
3. Specification and Testing of ARDs
4. Examples
5. Conclusion and Q&A



Context and Background

Challenges and innovations in clinical trial reporting

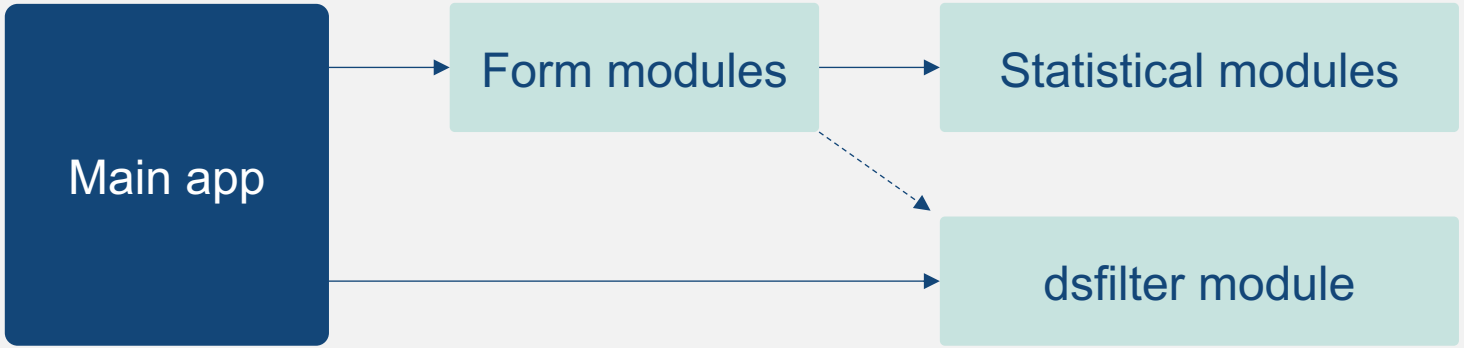
Definitions: business logic vs. service logic

Business logic: handles core computation, including data processing, rule enforcement, and (here) generating ARS-compliant datasets.

Service logic: transforms and formats the outputs from the business layer into TFLs. In the case of interactive apps, also orchestrates interactivity for end-users.

Background: Shiny app

Zafar Mohammad Sami, A. & Féat, Y. (2024). *R Shiny from Code to Cloud*. PHUSE EU Connect. https://phuse.s3.eu-central-1.amazonaws.com/Archive/2024/Connect/EU/Strasbourg/PAP_OS06.pdf



Current state and challenges of TFL production and QC in the context of clinical trial reporting

TFL production

- Custom script development
- Manual formatting & template adjustment

QC phase

- Manual cross-verification against analysis plans
- Ad-hoc QC & review tools
- Individual interpretation of requirements

Limited standardization & change tracking

- Variability according to individual expertise
- Custom, non-standarized data consolidation

Iterative re-checks & revisions

- Inconsistency in checks
- Extended review cycles
- Fragmented documentation
- Limited traceability

Current state and challenges of TFL production and QC in the context of clinical trial reporting

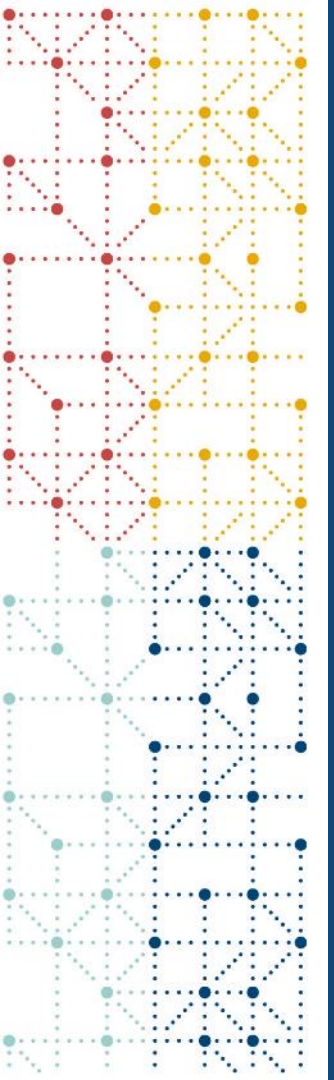
Critical limitations for interactive visualization and diverse data integration

Limited standardization & change tracking

- Variability according to individual expertise
- Custom, non-standarized data consolidation

Iterative re-checks & revisions

- Inconsistency in checks
- Extended review cycles
- Fragmented documentation
- Limited traceability



Introduction to ARS for TDD and their Benefits

Empowering consistency: the TDD and ARS advantage

Why TDD with ARS?

Streamlined development process

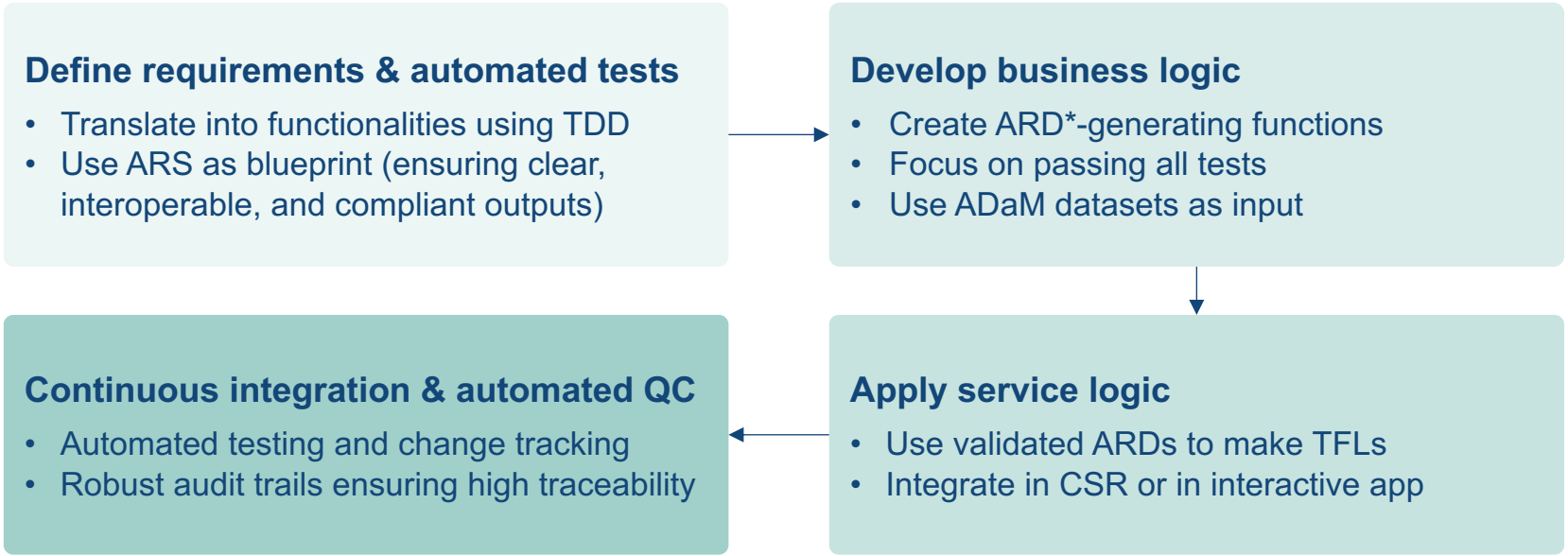
Ensuring that all requirements are met

Early QC

Enhanced collaboration through clear specifications for testing and production

Traceability and auditability

Unified TDD and ARS workflow



* Analysis Result Dataset (tabular representation of operation results in the ARS model)

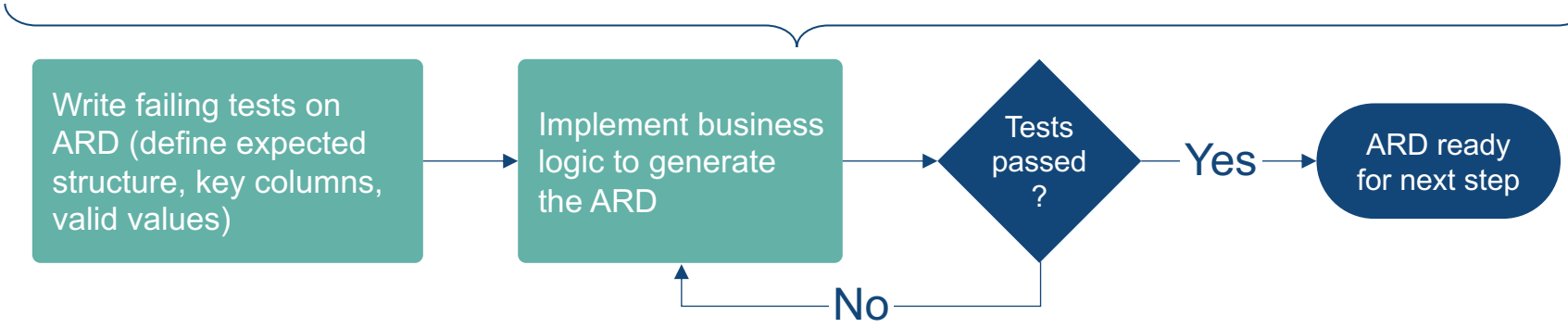
Unified TDD and ARS workflow

Define requirements & automated tests

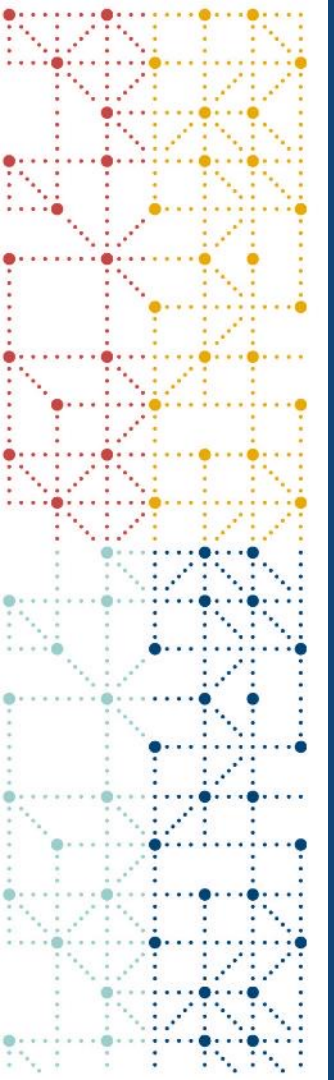
- Translate into functionalities using TDD
- Use ARS as blueprint (ensuring clear, interoperable, and compliant outputs)

Develop business logic

- Create ARD*-generating functions
- Focus on passing all tests
- Use ADaM datasets as input



* Analysis Result Dataset (tabular representation of operation results in the ARS model)

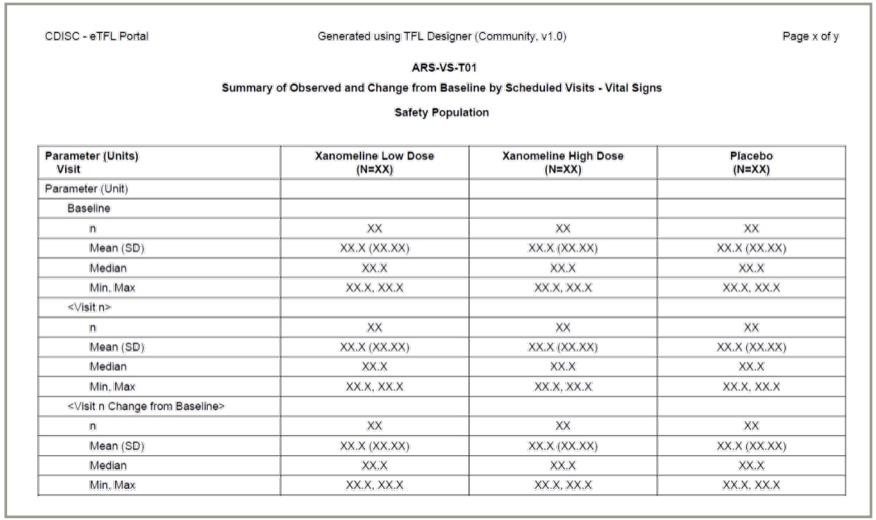


Specification and Testing of ARDs

Ensuring accuracy through pragmatic testing of ARDs

Prerequisite: specification of analysis methods

```
"name": "Summary by group of a continuous variable",
"id": "Mth02_ContVar_Summ_ByGrp",
"operations": [
  {
    "name": "Count of non-missing values",
    "id": "Mth02_ContVar_Summ_ByGrp_1_n",
    "label": "n",
    "resultPattern": "XX"
  },
  {
    "name": "Mean",
    "id": "Mth02_ContVar_Summ_ByGrp_2_Mean",
    "label": "Mean",
    "resultPattern": "XX.X"
  },
  {
    "name": "Standard deviation",
    "id": "Mth02_ContVar_Summ_ByGrp_3_SD",
    "label": "SD",
    "resultPattern": "(XX.XX)"
  },
```



CDISC - eTFL Portal | Generated using TFL Designer (Community, v1.0) | Page x of y

ARS-VS-T01

Summary of Observed and Change from Baseline by Scheduled Visits - Vital Signs

Safety Population

Parameter (Units) Visit	Xanomeline Low Dose (N=XX)	Xanomeline High Dose (N=XX)	Placebo (N=XX)
Parameter (Unit)			
Baseline			
n	XX	XX	XX
Mean (SD)	XX.X (XX.XX)	XX.X (XX.XX)	XX.X (XX.XX)
Median	XX.X	XX.X	XX.X
Min, Max	XX.X, XX.X	XX.X, XX.X	XX.X, XX.X
<Visit n>			
n	XX	XX	XX
Mean (SD)	XX.X (XX.XX)	XX.X (XX.XX)	XX.X (XX.XX)
Median	XX.X	XX.X	XX.X
Min, Max	XX.X, XX.X	XX.X, XX.X	XX.X, XX.X
<Visit n Change from Baseline>			
n	XX	XX	XX
Mean (SD)	XX.X (XX.XX)	XX.X (XX.XX)	XX.X (XX.XX)
Median	XX.X	XX.X	XX.X
Min, Max	XX.X, XX.X	XX.X, XX.X	XX.X, XX.X

"standalone specification of a set of 1 or more statistical operations that can be applied to any analysis variable to generate a set of analysis results"

Definition of ARDs on the CDISC Wiki

Pages / ... / OperationResult

Analysis Result Dataset (ARD)

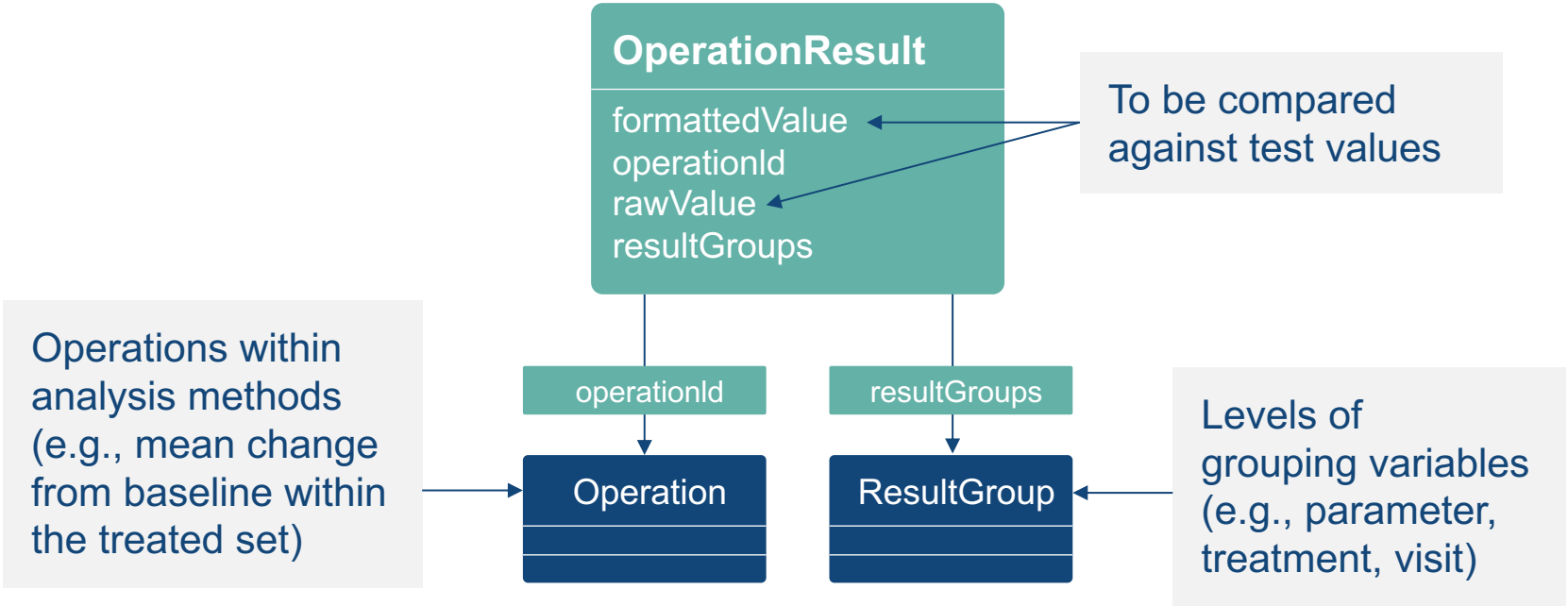
Created by Richard Marshall, last modified by Lorraine Sobson on Apr 18, 2024

When instances of the OperationResult class are represented in a tabular format, the resultant table may be referred to as an ARD if the following conditions are met:

- There is 1 row for each instance of the OperationResult class (i.e., 1 row per result value).
- Each row includes all the information needed to give context to the result value represented on the row, including at least:
  - the value of the **id** attribute of the analysis that generated and contains the result;
  - the value of the **operationId** attribute of the instance of the OperationResult class that is represented on the row; and
  - the values of all of the **groupingId**, **groupId**, or **groupValue** attributes that are present in **resultGroups** attribute of the instance of the OperationResult class that is represented on the row.

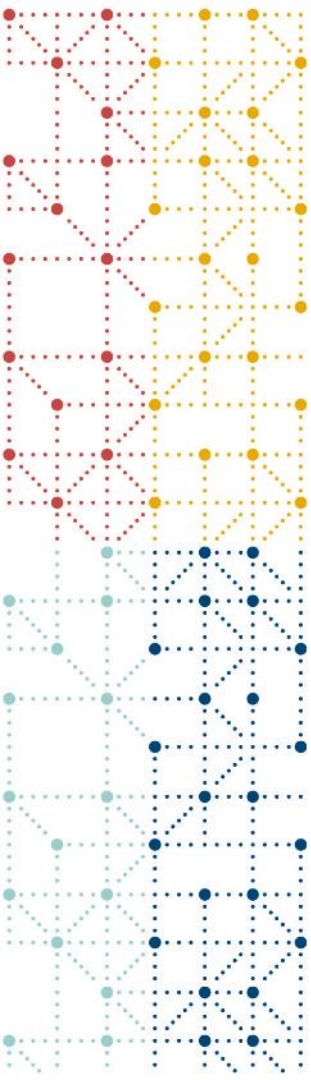
To aid readability, an ARD may include additional columns containing descriptive information that has been retrieved from instances of other classes using any of the identifier values included on each row. An ARD may contain results from 1 or more analyses. Each of the tables shown in the examples in the OperationResult section is an example of an ARD.

The OperationResult class



Types of unit tests for ARDs

Test type	Example	Purpose
Structure	Resulting ARD includes required sections: header, body, footer…	Confirm dataset adheres to the expected schema
Column & format	Contains key parameters (e.g., systolic BP, heart rate) in correct formats (e.g., number of decimal places)	Ensure all mandatory fields and formats are present
Calculation	The mean heart rate in an ARD generated using the 2025-04-04 ADaM snapshot matches the pre-calculated (using the same snapshot) value of 78 BPM.	Validate the correctness of computation results
Metadata compliance	Relevant metadata fields (e.g., data source) are correctly used.	Guarantee traceability and consistency
Consistency	Current ARD values are consistent with a fixed earlier snapshot, allowing for natural variation (e.g., the average HR at the current snapshot is within a 10% range from the reference 2025-04-04 ARD snapshot).	Detect regressions and unwanted changes in the data



Examples

Types of unit tests for ARDs

Test type	Example	Purpose
Structure	Resulting ARD includes required sections: header, body, footer…	Confirm dataset adheres to the expected schema
Column & format	Contains key parameters (e.g., systolic BP, heart rate) in correct formats (e.g., number of decimal places)	Ensure all mandatory fields and formats are present
Calculation	The mean heart rate in an ARD generated using the 2025-04-04 ADaM snapshot matches the pre-calculated (using the same snapshot) value of 78 BPM.	Validate the correctness of computation results
Metadata compliance	Relevant metadata fields (e.g., data source) are correctly used.	Guarantee traceability and consistency
Consistency	Current ARD values are consistent with a fixed earlier snapshot, allowing for natural variation (e.g., the average HR at the current snapshot is within a 10% range from the reference 2025-04-04 ARD snapshot).	Detect regressions and unwanted changes in the data

Example of individual unit test on an ARD

```r
# Use the ADVS dataset from an early ADaM snapshot for reproducibility
advs <- adam_snapshots[[ref_snapshot]]$advs
# Run the ARD-generating function to be tested
ard_mean_hr <- ard_vital_summary(advs)$Mean_Heart_Rate[1]
# Pre-calculated mean heart rate stored in a version-controlled file
expected_mean_hr <- test_values[[ref_snapshot]]$mean_heart_rate

# Verify that the ARD mean heart rate is as expected within tolerance
test_that("ARD mean heart rate matches expected mean within tolerance", {
  expect_true(abs(ard_mean - adam_mean) <= tolerance,
    info = sprintf(paste("ARD Mean Heart Rate = %.2f vs. expected %.2f exceeds",
      "the tolerance of %.4f"), ard_mean_hr, expected_mean_hr, tolerance)
  )
})
```

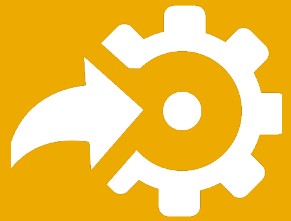
Pre-calculated test ARD

One ARD per analysis object, one unit test per row

operationName	resultGroup1_groupId	resultGroup1_groupLabel	resultGroup2_groupId	resultGroup2_groupLabel	resultGroup3_groupId	resultGroup3_groupLabel	rawValue	formattedValue
Count of non-missing values	AnlsGrouping_01_Trt_3	Xanomeline High Dose	AnlsGrouping_08_Param_1	Systolic Blood Pressure (mmHg)	AnlsGrouping_09_Visit_11	End of Treatment	218	218
Mean	AnlsGrouping_01_Trt_1	Placebo	AnlsGrouping_08_Param_4	Temperature (C)	AnlsGrouping_09_Visit_02	Week 2	-0.02454	-0.02
Median	AnlsGrouping_01_Trt_2	Xanomeline Low Dose	AnlsGrouping_08_Param_2	Diastolic Blood Pressure (mmHg)	AnlsGrouping_09_Visit_08	Week 20	-2	-2.0

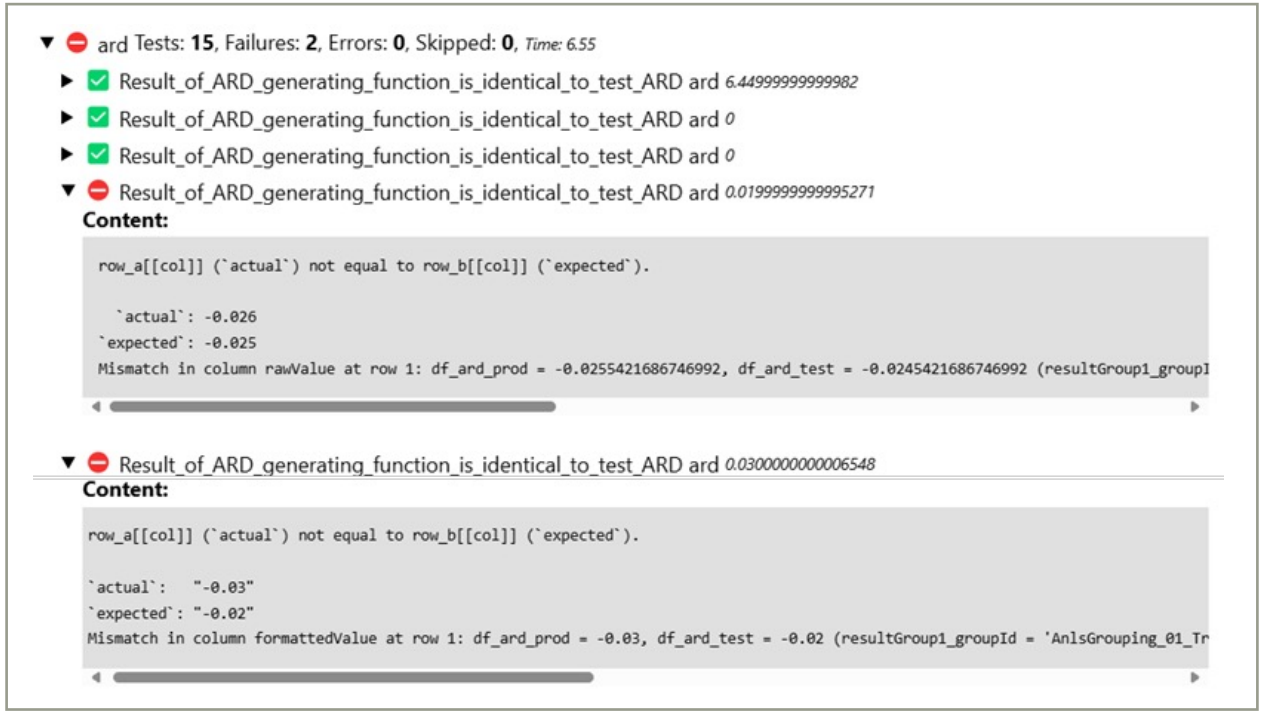
Central document(s) for TDD

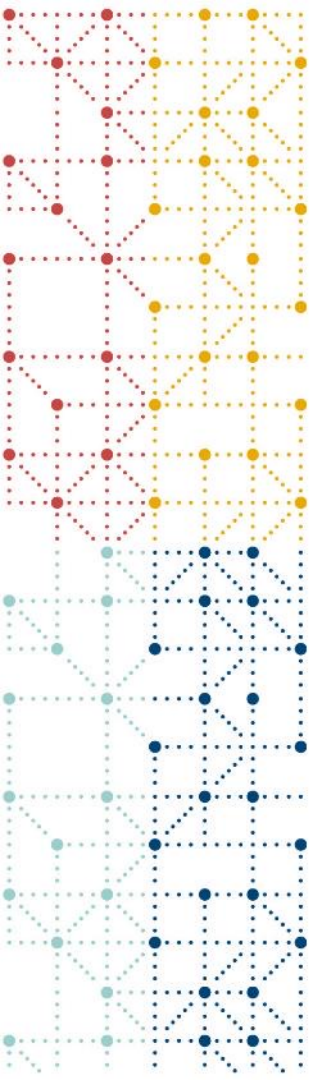
Such a test ARD with pre-calculated values facilitates all kinds of unit tests in an automated manner. It is self-documented, can be easily tracked, and enhances collaboration. It can be implemented using a different software from the main analysis (e.g., SAS vs. R).



Continuous integration

Sample output of unit tests suite run in a Cloud environment





Conclusion and Q&A

Efficiency:
standardization, separation of concerns (business vs. service), automation, reusable data

Collaboration:
enhanced teamwork, prospective planning, single source of truth

Traceability:
linking to protocol/SAP, consistency, transparency, standardized metadata

Thank You!

Yann Féat

Statistician
mainanalytics GmbH
Frankfurt am Main area, Germany
yann.feat@mainanalytics.de