

**sanofi**



# Automating SDTM and ADaM Creation in Clinical Trials

Lyubov Sushchenko, Sanofi, Bridgewater, USA  
Rie Ichihashi, Sanofi, Tokyo, Japan



# Table of Contents

- Introduction
- ADaM and the concept of the pseudo-code
- Nine Standard Derivation Types
- Automation of programs
- Benefits and challenges of pseudo-code



# INTRODUCTION

- Automating SDTM and ADaM in clinical trials
- Importance of the standards and specifications
- Concept of the pseudo-code and how it helps with automation

# ADAM AND THE CONCEPT OF THE PSEUDO-CODE

A specific way of writing the specifications to achieve automation.  
For each ADaM variable in ADaM metadata, a pseudo-code is created with standard derivation rules.

## Variable Metadata

| Dataset Name | Relative Order | Variable Name | Variable Label      | Type | Controlled Terminology | Source        | Derivation                                  | Assigned                     | Pseudo Code    | Core | Instructions for Programmers                                                                                  |
|--------------|----------------|---------------|---------------------|------|------------------------|---------------|---------------------------------------------|------------------------------|----------------|------|---------------------------------------------------------------------------------------------------------------|
| ADSL         | 170            | AAGEU         | Analysis Age Units  | text | AGEU                   |               |                                             | Assigned as "[YEARS, MONTHS, | assigned       | Cond | Conditionally required if AAGE is in                                                                          |
| ADSL         | 180            | AGEGR1        | Pooled Age Group 1  | text | [AGEGR1]               |               | For AGE of [xxx] to [xxx], AGEGR1 = [xxx]   |                              | recode         | Cond | Define AGE ranges in accordance with the study protocol. If AGE=, then AGEGR1=""                              |
| ADSL         | 190            | AGEGR2        | Pooled Age Group 2  | text | AGEGRPE                |               | For AGE of [xxx] to [xxx], AGEGR2 = [xxx]   |                              | condderivation | Cond | AGEGR2 is reserved for EudraCT : Preterm newborn - gestational age                                            |
| ADSL         | 210            | SEX           | Sex                 | text | SEX                    | DM.SEX        |                                             |                              | source         | Req  |                                                                                                               |
| ADSL         | 220            | RACE          | Race                | text | RACE                   | DM.RACE       |                                             |                              | source         | Req  | If the SDTM RACE is "MULTIPLE",                                                                               |
| ADSL         | 230            | RACEOR        | Original Race       | text | RACEC                  | SUPPDM.RACEOR |                                             |                              | source         | Perm |                                                                                                               |
| ADSL         | 240            | RACEGRy       | Pooled Race Group y | text | [RACEGRy]              |               | For RACE of [xxx] or [xxx], RACEGRy = [xxx] |                              | recode         | Perm | Character description of a grouping used for special case when study has RACE2, RACE3, for reporting purposes |

# ADAM AND THE CONCEPT OF THE PSEUDO-CODE

Pseudo-code is also applicable to value level derivations.

*WhereClauses, the main purpose is to provide metadata for VLM*

| Dataset Name | Variable Name | Where Variable | Comparator | Check Value | Value Label                     | Derivation                                                                                                        | Pseudo Code                                |
|--------------|---------------|----------------|------------|-------------|---------------------------------|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| ADVS         | AVAL          | PARAMCD        | EQ         | WEIGHT      | Weight (kg)                     | VS.VSSTRESN Where PARAMCD=VS.VSTESTCD                                                                             | copy_stres                                 |
| ADVS         | AVAL          | PARAMCD        | EQ         | DIABP       | Diastolic Blood Pressure (mmHg) | VS.VSSTRESN Where PARAMCD=VS.VSTESTCD. An average record is created for triplicate measurements on the same date. | copy_stres,<br>std_function(dtype_average) |
| ADVS         | AVAL          | PARAMCD        | EQ         | SYSBP       | Systolic Blood Pressure (mmHg)  | VS.VSSTRESN Where PARAMCD=VS.VSTESTCD. An average record is created for triplicate measurements on the same date. | copy_stres,<br>std_function(dtype_average) |

# NINE STANDARD DERIVATION TYPES

| Pseudo Code    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| source         | Indicates the variable has an origin of “Predecessor” where the source variable is defined in the “Source” column. This is used mainly for SDTM predecessor variables                                                                                                                                                                                                                                                                                             |
| assigned       | Assignment of a constant value for all the records in a dataset                                                                                                                                                                                                                                                                                                                                                                                                   |
| derivation     | Indicates that detailed derivation specs are available in the “Derivation” column in the “Variable Metadata” tab                                                                                                                                                                                                                                                                                                                                                  |
| recode         | Indicates that detailed derivation algorithm is available in the “recode” tab                                                                                                                                                                                                                                                                                                                                                                                     |
| rename         | Renaming a variable. This is used when a temporary variable is renamed into the ADaM dataset variable. The most common example is –DT, –DTM, and –TM variables. The ISO8601 datetime variables (e.g. EXSTDTC) in a source SDTM variable that is converted to temporary SAS date, datetime, and time variables (e.g. EXSTDTC, EXSTDTCM, and EXSTTM) early in the ADaM program, and renamed to suitable ADaM variable names (e.g. ADEX.ASTDT, ADSL.TRTSTDTCM, etc.) |
| condderivation | Indicates that detailed derivation algorithm is available in the “condderivation” tab                                                                                                                                                                                                                                                                                                                                                                             |
| lookup*        | Simple proc sql merge of 2 datasets, using subset condition if needed                                                                                                                                                                                                                                                                                                                                                                                             |
| std_function*  | Complex derivations which involve multiple datasets and/or multiple steps of computations that cannot be classified into the derivation types described above                                                                                                                                                                                                                                                                                                     |
| whereclauses*  | Indicates that pseudo-code for value level parameters is populated in the VLM table in the “WhereClauses” tab                                                                                                                                                                                                                                                                                                                                                     |

# NINE STANDARD DERIVATION TYPES

Three simple pseudo-codes “source”, “derivation”, and “assigned”. These pseudo-codes use the 3 corresponding variable metadata elements directly. Thus, the column which has the same name as the pseudo-code is used to pass the key element to the code generator.

*Columns for additional information for the code generator*

| Dataset Name | Variable Name | Variable Label                  | Type    | Source     | Derivation          | Assigned           | Pseudo Code |
|--------------|---------------|---------------------------------|---------|------------|---------------------|--------------------|-------------|
| ADSL         | STUDYID       | Study Identifier                | text    | DM.STUDYID |                     |                    | source      |
| ADSL         | TRTDURD       | Total Treatment Duration (Days) | integer |            | TRTEDT - TRTSDT + 1 |                    | derivation  |
| ADEG         | AWU           | Analysis Window Unit            | text    |            |                     | Assigned as "DAYS" | assigned    |

# NINE STANDARD DERIVATION TYPES

On the other hand, when the pseudo-code is "recode", "rename", "condderivation", "lookup", "std\_function" or "whereclauses", additional information needs to be entered in another tab that has the same name as the pseudo-code.

*Tabs for additional information for a code generator*

| Dataset Name       | Variable Name | Variable Label         | Type    | Source | Derivation                                 | Assigned              | Pseudo Code    | Core   | Instructions for Programmers                                                                                    |                      |
|--------------------|---------------|------------------------|---------|--------|--------------------------------------------|-----------------------|----------------|--------|-----------------------------------------------------------------------------------------------------------------|----------------------|
| ADAE               | AREL          | Analysis Causality     | text    |        | AE.AEREL. If AEREL is missing, AREL=[xxx]. |                       | condderivation | Perm   | For Pharma & Pasteur: If AEREL is derived due "Related" as per Safety Guidelines), AREL should be set to [xxx]. |                      |
| ADAE               | ARELN         | Analysis Causality (N) | integer |        |                                            | Numeric code for AREL | recode         | Perm   | The numeric code for AREL. One-to-one map to [xxx].                                                             |                      |
| ▶ Dataset Metadata |               | Variable Metadata      |         | recode |                                            | rename                | condderivation | lookup | std_function                                                                                                    | WhereClause: ... (+) |

# DETAILS ABOUT DERIVATION TYPES — “source”

1. Copy from a variable in the main source dataset (pseudo-code= “source”).

| Variables Example                          | Rendered SAS® code        |
|--------------------------------------------|---------------------------|
| ADSL.STUDYID, ADSL.USUBJID, ADSL.AGE, etc. | Keep STUDYID USUBJID AGE; |

*Pseudo-code “source” in the Variable Metadata tab*

| Dataset Name                                                                                                                                                               | Variable Name | Variable Label            | Type    | Source     | Derivation | Assigned | Pseudo Code |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|---------------------------|---------|------------|------------|----------|-------------|
| ADSL                                                                                                                                                                       | STUDYID       | Study Identifier          | text    | DM.STUDYID |            |          | source      |
| ADSL                                                                                                                                                                       | USUBJID       | Unique Subject Identifier | text    | DM.USUBJID |            |          | source      |
| ADSL                                                                                                                                                                       | AGE           | Age                       | integer | DM.AGE     |            |          | source      |
| <div><div>▶ ...</div><div>Variable Metadata</div><div>recode</div><div>rename</div><div>condderivation</div><div>lookup</div><div>std_function</div><div>... +</div></div> |               |                           |         |            |            |          |             |

# DETAILS ABOUT DERIVATION TYPES — “derivation”

2. Simple computation which appears in define.xml as the computational algorithm is directly used as the program code (pseudo-code = “derivation”).

| Variables Example                                  | Rendered SAS® code             |
|----------------------------------------------------|--------------------------------|
| Total Treatment Duration (Days)<br>(ADSL.TRTDURD). | TRTDURD = TRTEDT - TRTSDT + 1; |

*Pseudo-code “derivation” in the Variable Metadata tab*

| Dataset Name                                                                                                                                                                                         | Variable Name | Variable Label                  | Type    | Source | Derivation          | Assigned | Pseudo Code |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|---------------------------------|---------|--------|---------------------|----------|-------------|
| ADSL                                                                                                                                                                                                 | TRTDURD       | Total Treatment Duration (Days) | integer |        | TRTEDT - TRTSDT + 1 |          | derivation  |
| <div><div>▶ ...</div><div>Variable Metadata</div><div>recode</div><div>rename</div><div>condderivation</div><div>lookup</div><div>std_function</div><div>... (+)</div><div>⋮</div><div>◀</div></div> |               |                                 |         |        |                     |          |             |

# DETAILS ABOUT DERIVATION TYPES — “assigned”

3. Assignment of a constant value for all the records in a dataset (pseudo-code = “assigned”).

| Variables Example               | Rendered SAS® code |
|---------------------------------|--------------------|
| Analysis Window Unit (ADEG.AWU) | AWU = “DAYS”;      |

## *Pseudo-code “assigned” in the Variable Metadata tab*

| Dataset Name | Variable Name | Variable Label       | Type | Source | Derivation | Assigned           | Pseudo Code |
|--------------|---------------|----------------------|------|--------|------------|--------------------|-------------|
| ADEG         | AWU           | Analysis Window Unit | text |        |            | Assigned as "DAYS" | assigned    |

Variable Metadata recode rename condderivation lookup std\_function ... + ⋮ ◀

# DETAILS ABOUT DERIVATION TYPES — “recode”

4. Recode each value of a source variable to another value (pseudo-code = “recode”).

| Variables Example                                                         | Rendered SAS® code                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pooled Age Group 1 (ADSL.AGEGR1)<br>Pooled Age Group 1 (N) (ADSL.AGEGR1N) | <pre>if . &lt; AGE &lt; 60 then do;<br/>    AGEGR1 = "&lt; 60 years ";<br/>    AGEGR1N = 1;<br/>end;<br/>else if 60 &lt;= AGE &lt;= 70 then do;<br/>    AGEGR1 = "60-70 years";<br/>    AGEGR1N = 2;<br/>end;<br/>else if 70 &lt; AGE then do;<br/>    AGEGR1 = "over 70 years";<br/>    AGEGR1N = 3;<br/>end;</pre> |

# DETAILS ABOUT DERIVATION TYPES — “recode” (cont.)

## Pseudo-code “recode” in the Variable Metadata tab

| Dataset Name | Variable Name | Variable Label         | Type    | Source | Derivation        | Assigned                | Pseudo Code |
|--------------|---------------|------------------------|---------|--------|-------------------|-------------------------|-------------|
| ADSL         | AGEGR1        | Pooled Age Group 1     | text    |        | For AGE under 60, |                         | recode      |
| ADSL         | AGEGR1N       | Pooled Age Group 1 (N) | integer |        |                   | Numeric code for AGEGR1 | recode      |

► ... ata

Variable Metadata

recode

rename

condderivation

lookup

std\_function

Wl ...

+

⋮

◀

## Corresponding “recode” tab

| Datase t Name | Variable Name  | Source Variable | Source Variable Value | Output Value (Char) | Output Value (Num) |
|---------------|----------------|-----------------|-----------------------|---------------------|--------------------|
| ADSL          | AGEGR1 AGEGR1N | AGE             | (null:60)             | < 60 years          | 1                  |
| ADSL          | AGEGR1 AGEGR1N | AGE             | [60:70]               | 60-70 years         | 2                  |
| ADSL          | AGEGR1 AGEGR1N | AGE             | (70                   | over 70 years       | 3                  |

► ...

Variable Metadata

recode

rename

condderivation

lookup

std\_function

Wl

# DETAILS ABOUT DERIVATION TYPES — “rename”

5. Rename a variable (pseudo-code = “rename”).

This is used when a temporary variable is renamed in the ADaM dataset. The most common example is --DT, --DTM, and --TM variables.

| Variables Example                                                                                             | Rendered SAS® code                                                                         |
|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| Analysis Start Date (ADEX.ASTDT)<br>Analysis Start Datetime (ADEX.ASTDTM)<br>Analysis Start Time (ADEX.ASTTM) | <pre>rename  EXSTDT = ASTDT<br/>        EXSTDTM = ASTDTM<br/>        EXSTTM = ASTTM;</pre> |

# DETAILS ABOUT DERIVATION TYPES — “rename” (cont.)

## Pseudo-code “rename” in the Variable Metadata tab

| Dataset Name                                                                                     | Variable Name | Variable Label          | Type    | Source | Derivation      | Assigned | Pseudo Code                     |
|--------------------------------------------------------------------------------------------------|---------------|-------------------------|---------|--------|-----------------|----------|---------------------------------|
| ADEX                                                                                             | ASTDTM        | Analysis Start Datetime | integer |        | or EX.EXSTDTC   |          | dataset_derivation(DTC), rename |
| ADEX                                                                                             | ASTDT         | Analysis Start Date     | integer |        | date portion of |          | dataset_derivation(DTC), rename |
| ADEX                                                                                             | ASTTM         | Analysis Start Time     | integer |        | time portion of |          | dataset_derivation(DTC), rename |
| ▶ ... <b>Variable Metadata</b> recode rename condderivation lookup std_function Where( ... + : ◀ |               |                         |         |        |                 |          |                                 |

## Corresponding “rename” tab

| Dataset Name                                                         | Variable Name      | Source Variable Name   |
|----------------------------------------------------------------------|--------------------|------------------------|
| ADEX                                                                 | ASTDTM ASTDT ASTTM | EXSTDTC EXSTDTC EXSTTM |
| ◀ ▶ ... <b>Variable Metadata</b> recode <b>rename</b> condderivation |                    |                        |

# DETAILS ABOUT DERIVATION TYPES — “condderivation”

6. Derivation of a variable based on if-then conditions of the dependent variable(s) (pseudo-code = “condderivation”).

| Variables Example                                                                                                       | Rendered SAS® code                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Analysis Relative Day (ADY).<br>Derivations are different depends<br>whether ADT is before TRTSDT or<br>on/after TRTSDT | <pre>if . &lt; ADT &lt; TRTSDT then ADY = ADT - TRTSDT;<br/>else if ADT &gt;= TRTSDT &gt;. then ADY = ADT - TRTSDT + 1;</pre> |

# DETAILS ABOUT DERIVATION TYPES – “condderivation” (cont.)

## Pseudo-code “condderivation” in the Variable Metadata tab

| Dataset Name | Variable Name | Variable Label        | Type    | Source | Derivation                                                                                                         | Assigned | Pseudo Code    |
|--------------|---------------|-----------------------|---------|--------|--------------------------------------------------------------------------------------------------------------------|----------|----------------|
| ADEX         | ADY           | Analysis Relative Day | integer |        | [ADT-TRTSDT+1 if ADT is on or after the treatment start date, ADT-TRTSDT if ADT precedes the treatment start date] |          | condderivation |

► ...

Variable Metadata

recode

rename

condderivation

lookup

std\_function

Where...

+

◀

## Corresponding “condderivation” tab

| Dataset Name | Variable Name | Derivation Condition | Output Variable Value | Source Variable or Formula |
|--------------|---------------|----------------------|-----------------------|----------------------------|
| ADEX         | ADY           | . < ADT < TRTSDT     |                       | ADT - TRTSDT               |
| ADEX         | ADY           | ADT >= TRTSDT > .    |                       | ADT - TRTSDT + 1           |

►

Dataset Metadata

Variable Metadata

recode

rename

condderivation

lookup

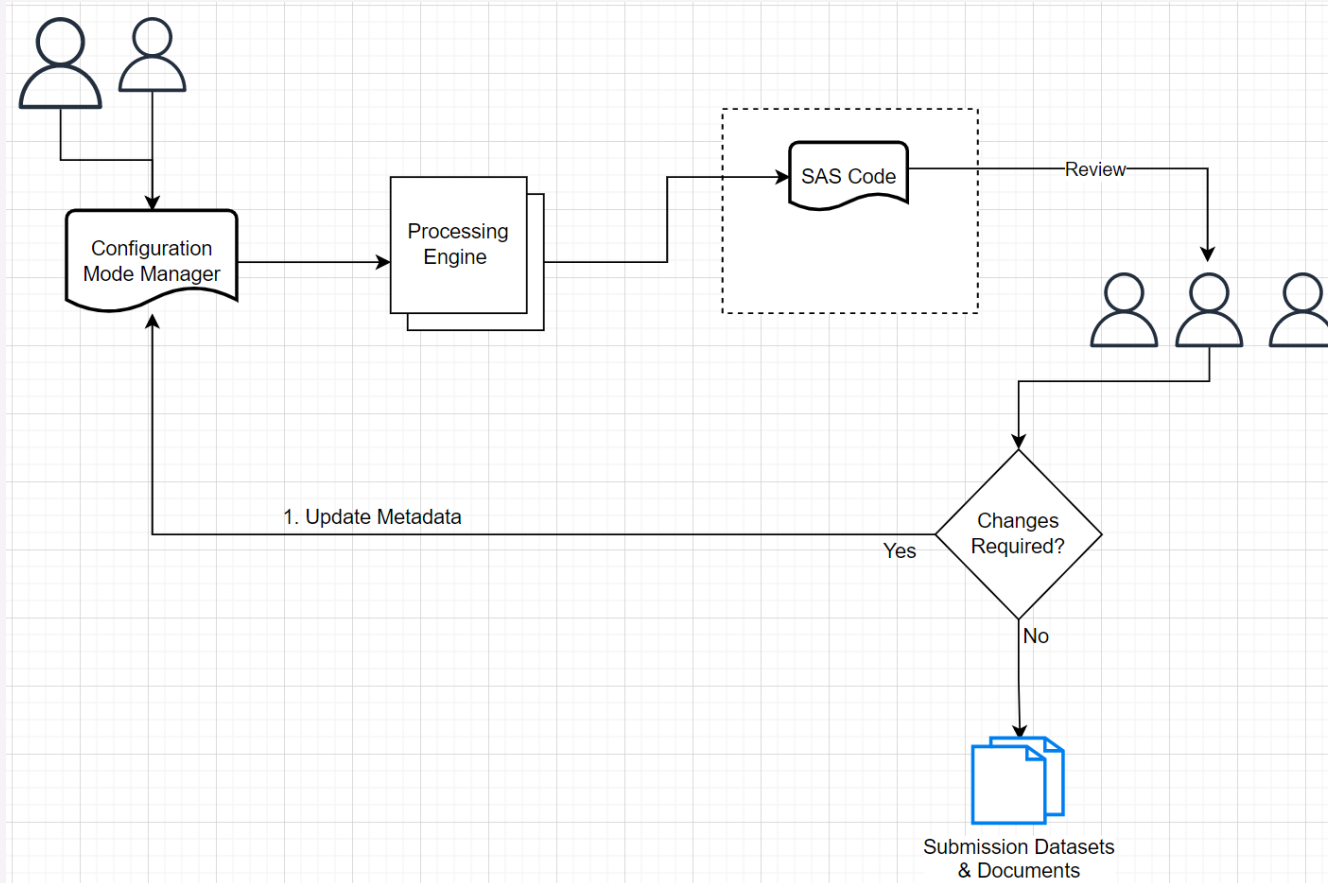
std\_function

W

# AUTOMATION OF PROGRAMS

- To achieve automation, we are taking a configuration-based approach
- To derive a variable, we created a generic configuration model, that captures all necessary information. The code generator is utilizing this configuration along with other parameters (program formats etc.) to generate output code.
- A code generator is a software tool that automatically generates code based on input specifications or parameters.

# AUTOMATION OF PROGRAMS



# BENEFITS AND CHALLENGES OF PSEUDO-CODE

Even with the project still in development, these are the proposed benefits of the pseudo-codes:

- **Agnostic**

Pseudo-code is language agnostic.

- **Consistent**

Pseudo-code and additional user input in corresponding columns and tabs serve as the source for code generator to generate the program as well as the derivation specifications for the programmers.

- **Structured**

Pseudo-code provides a structure to the definition of derivation algorithms.

# BENEFITS AND CHALLENGES OF PSEUDO-CODE

We tested out code generator on several pilot studies, these were the challenges we faced:

- Pseudo-code and code generator functions are new concepts and need user training and support.
- Could be challenging at times to define correct/complicated pseudo-codes at the study level.
- This process adds a non-trivial layer of metadata which requires modern tools to be able to preview future SAS codes and enter the pseudo-codes.

# ACKNOWLEDGEMENT

The authors of this paper would like to express their gratitude to the Sanofi ART team, for their support and ideas, especially Faisal Mohammed. ADaM MAP team for their dedication and effort. B&P programmers for providing feedback and volunteering in testing out new tools. And Beilei Xu for inspiring to write this paper.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged.

### Contact the authors at:

Author Name: Lyubov Sushchenko  
Company: Sanofi US  
Address: 55 Corporate Drive  
City / Postcode: Bridgewater, 08807  
Email: [Lyubov.Sushchenko@sanofi.com](mailto:Lyubov.Sushchenko@sanofi.com)  
Web: <https://www.sanofi.com>

Author Name: Rie Ichihashi  
Company: Sanofi K.K.  
Address: Tokyo Opera City Tower, 3-20-3, Nishi Shinjuku  
City / Postcode: Tokyo 163-1488, Japan  
Email: [Rie.Ichihashi@sanofi.com](mailto:Rie.Ichihashi@sanofi.com)  
Web: <https://www.sanofi.com>

---

Brand and product names are trademarks of their respective companies.



**sanofi**