



APAC 2026

19–21 February

Novotel Hyderabad International Convention Centre

**FROM DOCUMENT TO DERIVATION: AUTOMATING ADAM VARIABLE
AND ENDPOINT CODE GENERATION USING STUDY SPECIFICATIONS
AND CLINICAL DOCUMENTS**

**Alisha D, Sruthy Biju
Pfizer, India**



DISCLAIMER

I confirm that, the opinions and thoughts discussed in this presentation are subject to my own independent views and are not subject to the opinions of the organization that I represent.



AGENDA

- 1 INTRODUCTION**
- 2 OBJECTIVES OF FRAMEWORK**
- 3 OVERVIEW OF MODELS & ALGORITHMS**
- 4 VARIABLE METADATA → R CODE GENERATION WORKFLOW**
- 5 ENDPOINT REPOSITORY & RETRIVAL WORKFLOW**
- 6 USER INTERFACE & DASHBOARD DEMONSTRATION**
- 7 COMPARISON EVALUATION WITH OTHER MODELS**
- 8 PERFORMANCE EVALUATION**
- 9 FUTURE ENHANCEMENTS**
- 10 CONCLUSION**



INTRODUCTION

- Translating clinical dataset specifications into executable derivation code is typically a manual, interpretation-heavy task, often hindered by complex logic and evolving rules. To streamline this, we introduce an offline, automated framework that converts derivation descriptions into validated, reproducible R code, handling mappings, derived variables, and optimized logic.
- The system uses a locally deployed code-generation model with rule-aware preprocessing, ensuring traceability, repeatability, and alignment with CDISC standards. It also offers controlled endpoint support for rule retrieval and modification, serving as a supplementary capability.



OBJECTIVES OF FRAMEWORK

- Automate the conversion of clinical dataset specifications into executable R code.
- Ensure reproducibility and traceability of endpoint derivations.
- Handle derivation logic, dataset dependencies, and optimized mutation logic.
- Ensure alignment with CDISC standards throughout the process.

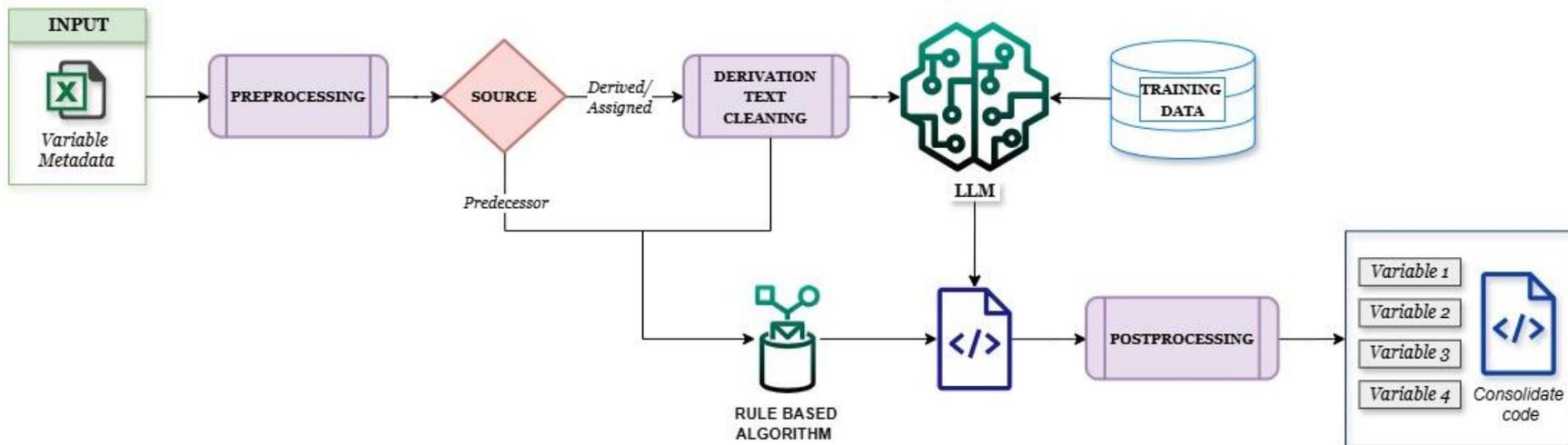


OVERVIEW OF MODELS & ALGORITHMS

COMPONENT	PURPOSE	KEY FEATURES
CodeGen-350M-Multi	Generate R code for derived variables	Multilingual training, fine-tuned on CDISC patterns
Sentence Transformer	Create embeddings for endpoint definitions	Dense semantic representation, cosine similarity
FAISS Index	Efficient retrieval of similar endpoints	Inner-product search, scalable indexing
Rule-Based Engine	Direct mapping of predecessor variables	Deterministic logic, reproducibility

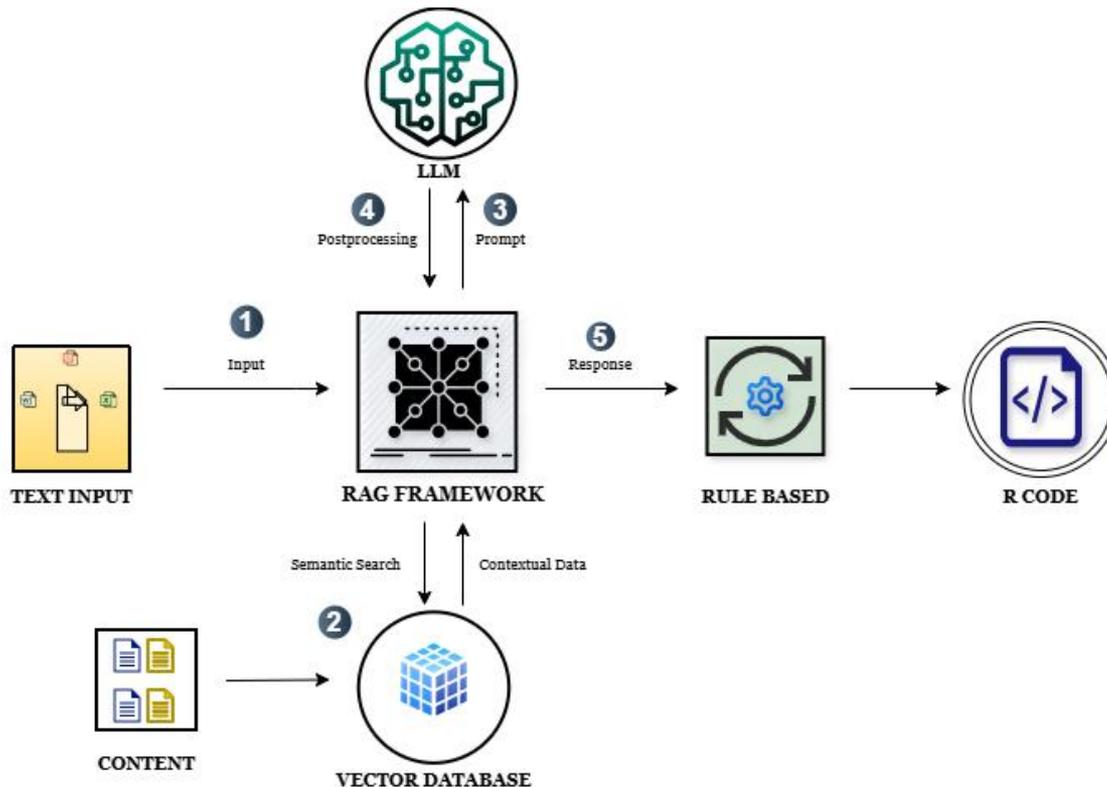


VARIABLE METADATA → R CODE GENERATION WORKFLOW





ENDPOINT REPOSITORY & RETRIVAL WORKFLOW





USER INTERFACE & DASHBOARD DEMONSTRATION

A screenshot of a web application interface. At the top right, there is a status indicator showing a person icon, the text "RUNNING...", and two buttons labeled "Stop" and "Deploy". The main heading is "DoC to DERV" with a colorful starburst icon to its left. Below this, there are two navigation links: "R Code Generator (Spec -> R)" which is underlined in red, and "Endpoint Rule -> R Code". A horizontal line separates this from the next section, "R Code Generator from Spec Excel", which has a colorful bar chart icon. Underneath is a sub-heading "Upload Dataset Specification Excel" with a folder icon. A large light gray box contains a cloud icon with a plus sign, the text "Drag and drop file here", and "Limit 200MB per file • XLSX". To the right of this box is a "Browse files" button.



USER INTERFACE & DASHBOARD DEMONSTRATION

DoC to DERV

R Code Generator (Spec -> R) Endpoint Rule -> R Code

R Code Generator from Spec Excel

Upload Dataset Specification Excel

Drag and drop file here
Limit 200MB per file • XLSX

ADSL_Endpoint_Spec.xlsx 9.5KB

Generating R code...

ADSL Variable Metadata

Variable Name	Variable Label	Type	Codelist/Controlled Terms	Source/Derivation/Comment
USUBJID	Unique Subject Identifier	text		Predecessor: DM.USUBJID
DTHDT	Date of Death	integer		Derived: DM.DTHDTC converted from ISO to numeric
DTHCAUS	Cause of Death	text		Derived: DD.DDORRES where DD.DDTESTCD is 'PRCDTH'.
EOSDT	End of Study Date	integer		Derived: DS.DSSTDTC where DS.DSCAT = 'DISPOSITION EVENT' then converted to numeric.
EOSST	End of Study Status	text	COMPLETED; DISCONTINUED	Derived: if DS.DSDECOD = 'COMPLETED' where DS.DSCAT = 'DISPOSITION EVENT' set to 'COMPLETED' if DS.DSDECOD = 'LOST TO FOLLOW-UP' where DS.DSCAT = 'DISPOSITION EVENT' set to 'DISCONTINUED' if DS.DSDECOD = 'DEATH' where DS.DSCAT = 'DISPOSITION EVENT' set to 'COMPLETED'
COMPLFL	Completed Population Flag	text	Y	Assigned: Set to Y where EOSST = 'COMPLETED' Otherwise null
TRTSDT	Date of First Exposure to Treatment	integer		Predecessor: DM.RFXSTDTC converted from ISO formatted text to numeric



USER INTERFACE & DASHBOARD DEMONSTRATION

DoC to DERV

[R Code Generator \(Spec → R\)](#) | [Endpoint Rule → R Code](#)

R Code Generator from Spec Excel

Upload Dataset Specification Excel

Drag and drop file here
Limit 200MB per file • XLSX Browse files

ADSI_Endpoint_Spec.xlsx 9.5KB ×

Code generated successfully

Select Variable Name
COMPLFL ▼

Derivation

Derivation

Set to Y where EOSSTT = 'COMPLETED' Otherwise null

R Code

```
library(dplyr)
data <- data %>%
  mutate(COMPLFL = case_when(EOSSTT == "COMPLETED" ~ "Y",
                             TRUE ~ NA_character_))
```



USER INTERFACE & DASHBOARD DEMONSTRATION

The screenshot displays the 'DoC to DERV' dashboard. At the top, there are navigation links for 'R Code Generator (Spec + R)' and 'Endpoint Rule + R Code'. The main section is titled 'R Code Generator from Spec Excel' and includes an 'Upload Dataset Specification Excel' step. A file upload area shows a file named 'ADSL_Endpoint_Spec.xlsx' (9.5KB) has been successfully uploaded. Below this, a green notification bar states 'Code generated successfully'. A 'Select Variable Name' dropdown menu is open, showing a list of variables: 'COMPLFL', 'DTHCAUS', 'DTHDT', 'EOSDT', 'EOSSTT', 'TRTSDT', and 'USUBJID'. The 'COMPLFL' variable is currently selected.



USER INTERFACE & DASHBOARD DEMONSTRATION

DoC to DERV

[R Code Generator \(Spec + R\)](#) [Endpoint Rule + R Code](#)

R Code Generator from Spec Excel

Upload Dataset Specification Excel

Drag and drop file here
Limit 200MB per file • XLSX Browse files

ADSL_Endpoint_Spec.xlsx 0.5KB ×

Code generated successfully

Select Variable Name
EOSSTT ▾

Derivation

Derivation

```
if DS.DSDECOD = 'COMPLETED' where DS.DSCAT = 'DISPOSITION EVENT' set to 'COMPLETED' if DS.DSDECOD = 'LOST TO FOLLOW-UP' where DS.DSCAT = 'DISPOSITION EVENT' set to 'DISCONTINUED' if DS.DSDECOD = 'DEATH' where DS.DSCAT = 'DISPOSITION EVENT' set to 'COMPLETED'
```

R Code

```
library(dplyr)
ds <- ds %>%
  mutate(EOSSTT = case_when(DSCAT == "DISPOSITION EVENT" & DSDECOD == "COMPLETED" ~ "COMPLETED",
    DSCAT == "DISPOSITION EVENT" & DSDECOD == "LOST TO FOLLOW-UP" ~ "DISCONTINUED",
    DSCAT == "DISPOSITION EVENT" & DSDECOD == "DEATH" ~ "COMPLETED",
    TRUE ~ NA_character_))
```



USER INTERFACE & DASHBOARD DEMONSTRATION

[Generate Final R Script](#)

```
library(dplyr)
library(admiral)

dd <- dd %>%
  mutate(DTHCAUS = if_else(DOTESTCD == "PRCDTH",
    DDOORRES,
    NA_character_))

ds <- ds %>%
  mutate(EOSSTT = case_when(DSCAT == "DISPOSITION EVENT" & DSDECOD == "COMPLETED" ~ "COMPLETED",
    DSCAT == "DISPOSITION EVENT" & DSDECOD == "LOST TO FOLLOW-UP" ~ "DISCONTINUED",
    DSCAT == "DISPOSITION EVENT" & DSDECOD == "DEATH" ~ "COMPLETED",
    TRUE ~ NA_character_) %>%
  mutate(COMPLFL = case_when(EOSSTT == "COMPLETED" ~ "Y",
    TRUE ~ NA_character_))

dm <- dm %>%
  derive_vars_dt(
    new_vars_prefix = "DTH",
    dtc = DTHDTC
  )

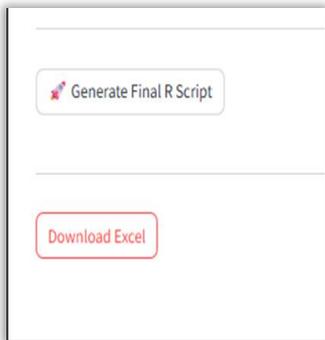
dm <- dm %>%
  derive_vars_dt(
    new_vars_prefix = "TRTS",
    dtc = RFXSTDTC
  )

ds <- ds %>%
  derive_vars_dt(
    new_vars_prefix = "DSST",
    dtc = DSSTDTC
  ) %>%
  mutate(EOSDT = if_else(DSCAT == "DISPOSITION EVENT",
    DSSTDTC,
    NA_character_))
```

[Download Excel](#)



USER INTERFACE & DASHBOARD DEMONSTRATION



	A	B	C	D	E	F
	Variable Name	Variable Label	Type	Codelist/Controlled Terms	Source/Derivation/Comment	R_code
1						
2	USUBJID	Unique Subject Identifier	text		Predecessor: DM.USUBJID	
3	DTHDT	Date of Death	integer		Derived: DM.DTHDTC converted from ISO to numeric	<pre>dm <- dm %>% derive_vars_dt(new_vars_prefix = "DTH", dtc = DTHDTC)</pre>
4	DTHCAUS	Cause of Death	text		Derived: DD.DDORRES where DD.DDTESTCD is 'PRCDTH'.	<pre>dd <- dd %>% mutate(DTHCAUS = if_else(DDTESTCD == "PRCDTH", DDORRES, NA_character_))</pre>
5	EOSDT	End of Study Date	integer		Derived: DS.DSSTDTC where DS.DSCAT is 'DISPOSITION EVENT' then converted to numeric.	<pre>ds <- ds %>% derive_vars_dt(new_vars_prefix = "DSST", dtc = DSSTDTC) %>% derive_vars_dt(new_vars_prefix = "DSST", dtc = DSSTDTC) %>% mutate(EOSDT = if_else(DSCAT == "DISPOSITION EVENT", DSSTDTC, NA_character_))</pre>



USER INTERFACE & DASHBOARD DEMONSTRATION

DoC to DERV

 R Code Generator (Spec → R)  Endpoint Rule → R Code

Endpoint Rule → R Code Generator

Enter your requirement:

Describe your business rule

 Generate Rule & R Code

Enter text and click Generate to proceed.



DoC to DERV

 R Code Generator (Spec → R)

 Endpoint Rule → R Code

Endpoint Rule → R Code Generator

Enter your requirement:

Overall Survival is the number of days from treatment start date to death from any cause

 [Generate Rule & R Code](#)

Enter text and click Generate to proceed.



USER INTERFACE & DASHBOARD DEMONSTRATION

Controlled Rule Editor

Base Rule

Assign Overall Survival (OS) as the number of days from treatment start date (TRT_STDT) to date of death (DTHDT).
If death occurs after treatment start, set ADT to DTHDT and set censoring indicator CNSR to 0.
If death does not occur or occurs before treatment start, censor the subject by setting ADT to last known survival contact date (SRVLACDT) and set CNSR to 1.
Calculate AVAL as ADT minus treatment start date plus 1, with a minimum value of 1 day

Editable Fields (CAPS only)

ADT	AVAL	CNSR
<input type="text" value="ADT"/>	<input type="text" value="AVAL"/>	<input type="text" value="CNSR"/>
DTHDT	SRVLACDT	TRT_STDT
<input type="text" value="DTHDT"/>	<input type="text" value="SRVLACDT"/>	<input type="text" value="TRT_STDT"/>

+ Append Approved Clause

Select a clause to append

None

Final Rule

Assign Overall Survival (OS) as the number of days from treatment start date (TRT_STDT) to date of death (DTHDT).
If death occurs after treatment start, set ADT to DTHDT and set censoring indicator CNSR to 0.
If death does not occur or occurs before treatment start, censor the subject by setting ADT to last known survival contact date (SRVLACDT) and set CNSR to 1.
Calculate AVAL as ADT minus treatment start date plus 1, with a minimum value of 1 day

Generated R Code

```
mutate(  
  ADT = case_when(  
    !is.na(DTHDT) & DTHDT >= TRT_STDT ~ DTHDT,  
    TRUE ~ SRVLACDT  
  ),  
  CNSR = case_when(  
    !is.na(DTHDT) & DTHDT >= TRT_STDT ~ 0,  
    TRUE ~ 1  
  ),  
  AVAL = pmax(as.numeric(ADT - TRT_STDT + 1), 1)  
)
```



USER INTERFACE & DASHBOARD DEMONSTRATION

AVAL	CNSR
AVAL	CNSR
SRVLACDT	TRT_STDT
SRVLACDT	ARMCD



USER INTERFACE & DASHBOARD DEMONSTRATION

+ Append Approved Clause

Select a clause to append

None

Apply Rule Changes

Final Rule

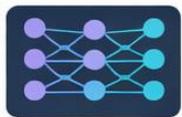
Assign Overall Survival (OS) as the number of days from treatment start date (ARMCD) to date of death (DTHDT).
If death occurs after treatment start, set ADT to DTHDT and set censoring indicator CNSR to 0.
If death does not occur or occurs before treatment start, censor the subject by setting ADT to last known survival contact date (SRVLACDT).
Calculate AVAL as ADT minus treatment start date plus 1, with a minimum value of 1 day

Generated R Code

```
mutate(  
  ADT = case_when(  
    !is.na(DTHDT) & DTHDT = ARMCD ~ DTHDT,  
    TRUE ~ SRVLACDT  
  ),  
  CNSR = case_when(  
    !is.na(DTHDT) & DTHDT >= ARMCD ~ 0,  
    TRUE ~ 1  
  ),  
  AVAL = pmax(as.numeric(ADT - ARMCD + 1), 1)  
)
```

COMPARISON EVALUATION WITH OTHER MODELS

CodeGen-350M-Multi



CODEGEN

- ✓ Balanced Performance
- ✓ Accurate & Efficient
- ✓ Low Resource Usage

Salesforce / CodeT5-Base



- ✓ Faster Generation
- ✓ But Less Accurate
- ✓ Lower Quality Code

CodeLlama-7B



- ✗ High Quality Code
- ✗ But Slower Inference
- ✗ High Resource Usage

Key Takeaway:

CodeGen-350M-Multi: Optimal Balance of Speed and Accuracy

Effective R Code Derivation with Reduced Computational Demands



PERFORMANCE EVALUATION

Code Generation Accuracy



70%

Fully Functional R Code Generated

- ▶ Aligned with Reference Logic
- ▶ Minor Errors Due to Limited Data

Endpoint Retrieval Efficiency



MRR: 0.80

Mean Reciprocal Rank

- ▶ High Accuracy in Top Results
- ▶ Effective Endpoint Retrieval

Reliable Performance in Code Generation and Endpoint Retrieval

Ensuring ADaM-Compliant Workflow



FUTURE ENHANCEMENTS



Expand training to all BDS data structures



Handle complex post-processing scenarios



Enable advanced derivation logic



Integrate validation, QC, and traceability



Support adaptive learning for improvement



CONCLUSION: KEY TAKEAWAYS

1. Framework for R Code Generation

- Automates translation of clinical documents into executable R code, reducing manual effort.
- Ensures standardization and traceability aligned with **CDISC ADaM** principles.

2. Hybrid Automation Approach

- Combines deterministic rules and a lightweight code generation model.
- Balances automation with regulatory compliance and auditability.

3. Governance with Customization

- Separates rule governance from code generation to prevent logic drift.
- Allows study-specific customization within clear boundaries.



REFERENCES

- Jiang, J., Wang, F., et al. (2024). A Survey on Large Language Models for Code Generation. arXiv preprint.
- Nijkamp, E., Pang, B., Hayashi, H., et al. (2022). CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis. arXiv preprint.
- Tenneti, L. S. D., Gayatri, M. K., Prabha, M. D., & Shravani, D. (2025). Code Generation Using LLMs. Future Engineering Journal.
- E. Dehaerne, B. Dey, S. Halder, S. De Gendt and W. Meert, "Code Generation Using Machine Learning: A Systematic Review," in IEEE Access, vol. 10, pp. 82434-82455, 2022, doi: 10.1109/ACCESS.2022.3196347.
- Nijkamp, Erik, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. "A Conversational Paradigm for Program Synthesis." arXiv preprint (2022).
- CDISC. (2026, Jan 20). CDISC standards. Clinical Data Interchange Standards Consortium. <https://www.cdisc.org/>



THANK YOU