

Single Day Event

Get It Done: Accelerating Clinical Trial Submissions with SAS Agents

Streamlining the path to regulatory submission with SAS Viya

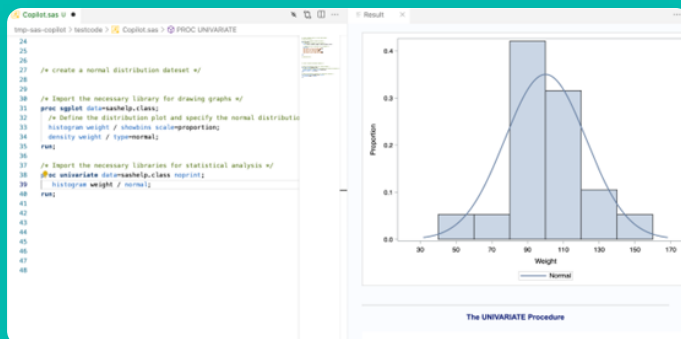
Soundarya Palanisamy, Sr Industry Advisor
Global Health and Life Sciences Practice
SAS Institute

phuse SAS and Generative AI (GenAI)



■ SAS Viya Copilot to accelerate user productivity

- Conversationally query SAS Viya to increase user productivity, further democratizing analytics

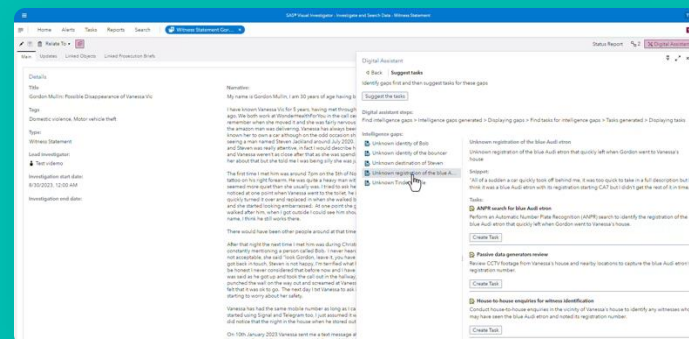


example: SAS Code Generation



■ Specialized SAS Viya Copilots to accelerate industry workflows and processes

- GenAI driven assistants to accelerate domain-specific workflows

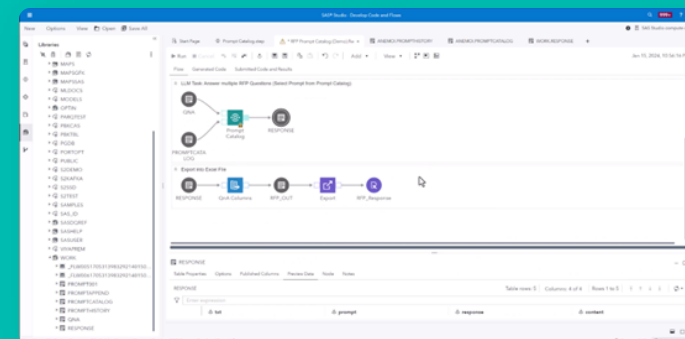


example. Clinical Data Lifecycle



■ SAS Viya to help build your LLM-based applications to unlock value in your proprietary business assets

- Flexible SAS Viya GenAI building blocks to help with development and orchestration of your LLM application stack



example. Build Prompt Library using existing SAS Viya APIs and SAS Studio steps



■ Standalone GenAI services to enable your toolchains, like a synthetic data generation service

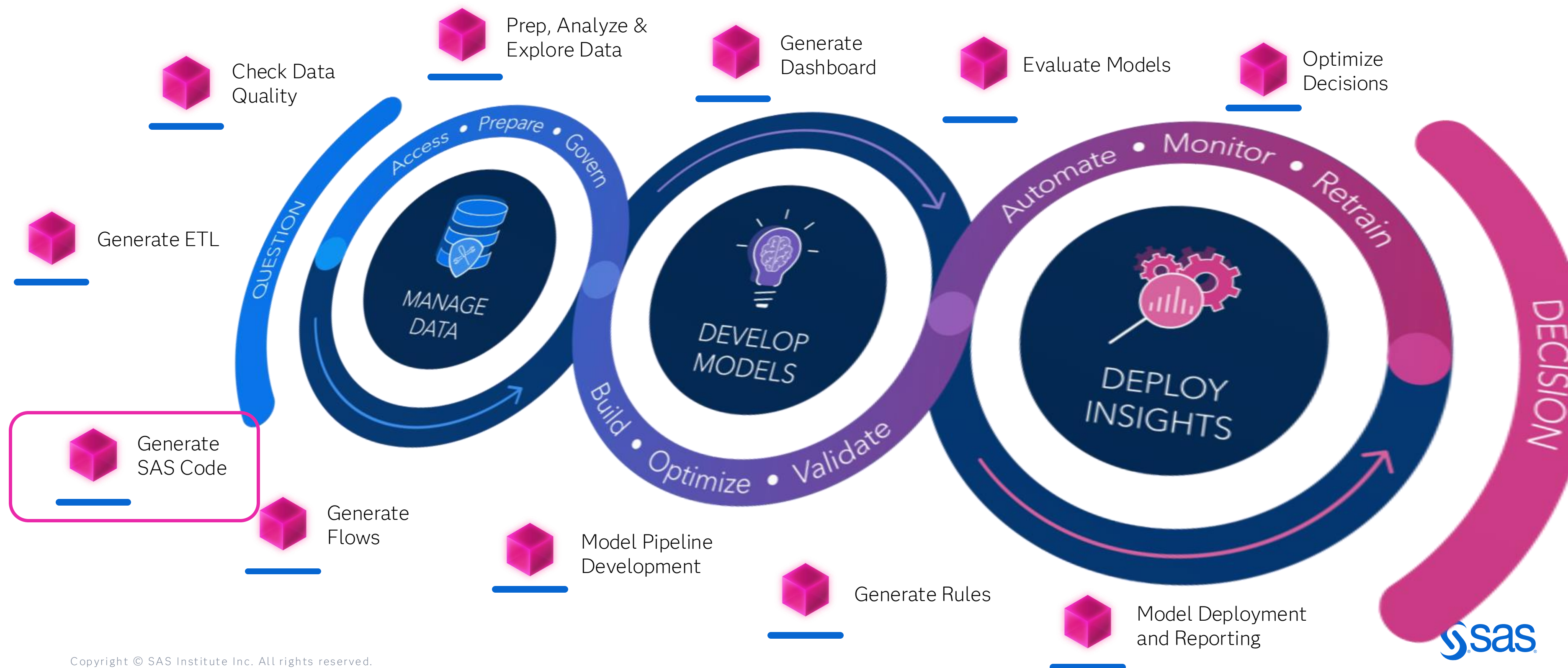
- API-first low-code/no-code synthetic data generation and validation experience



example. Synthetic data generation API

Your AI assistant for the Data and AI Lifecycle

Ease the barrier to data analysis and AI development





SAS Code Generation

Code Generation.sas

```
8 data mycas.pharma;  
33 25 1 07 31 0  
34 24 0 48 39 0  
35 25 1 56 50 0  
36 ;  
37 run;  
38  
39  
40 /* print the pharma dataset */  
41 proc print data=mycas.pharma;  
42 run;  
43  
44 /* Create a frequency table on occurred */  
45 proc freq data=mycas.pharma;  
46 | tables Event_Occurred;  
47 run;  
48  
49 /* Generate a scatter plot of Age_at_Study_Entry against Time_to_Event grouped by Event_Occurred */  
proc sgplot data=mycas.pharma;  
scatter x=Age_at_Study_Entry y=Time_to_Event / group=Event_Occurred;  
run;  
50  
51  
52  
53 /* Estimate a Kaplan-Meier Survival Function for Time_to_Event and Event_Occurred by Treatment_Group */  
54  
55  
56 /* Estimate a proportional hazard model for Time_to_Event and Event_Occurred against Treatment_Group and Age_at_Study_Entry */  
57  
58  
59  
60
```

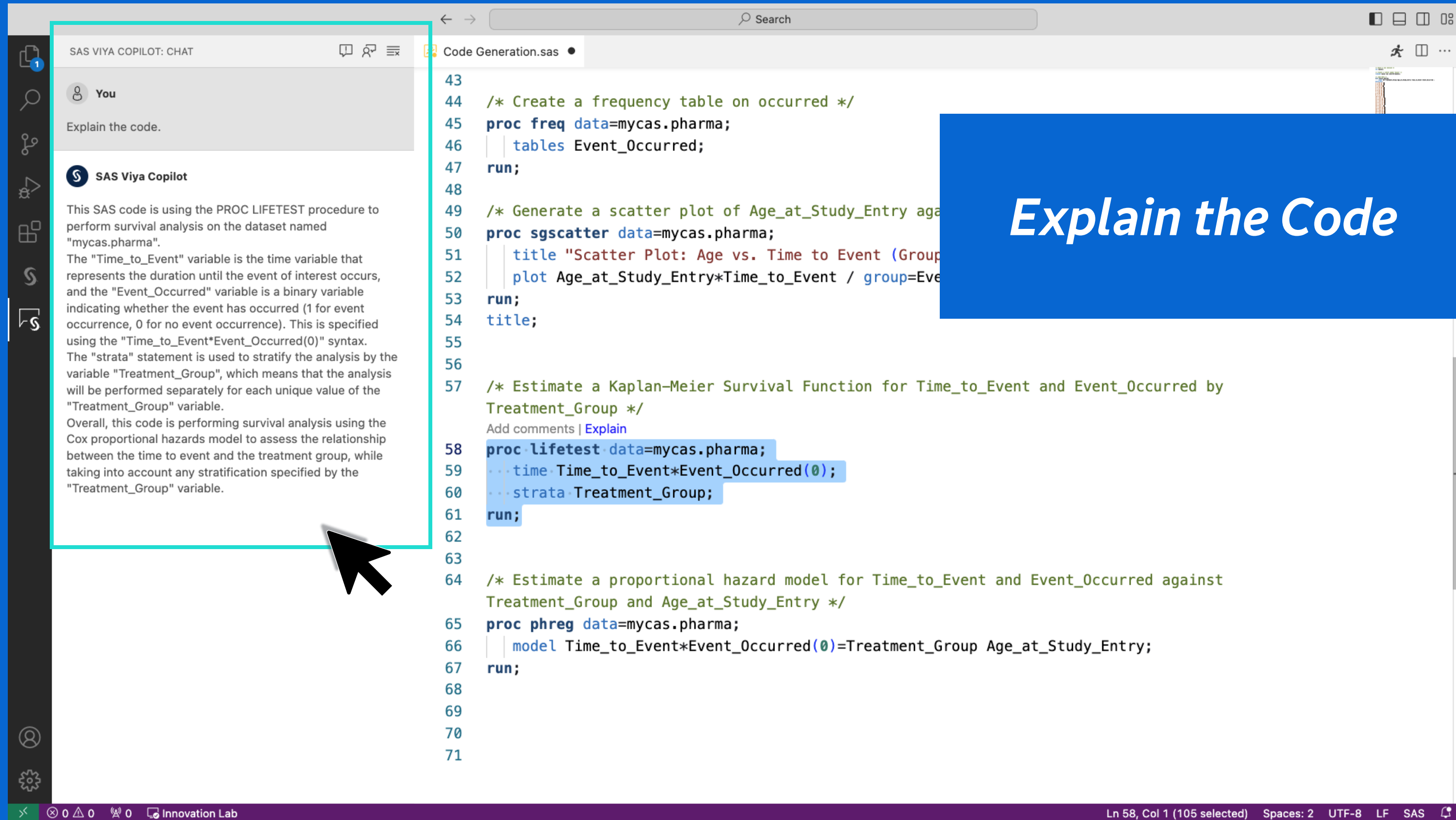
Generate code

< 1/1 > Accept Tab Accept Word ⌘ → ...

Ln 49, Col 100 Spaces: 2 UTF-8 LF SAS

Generate Code

SAS Code Generation



The screenshot displays the SAS Viya Copilot interface. On the left, a chat window titled "SAS VIYA COPILOT: CHAT" shows a conversation. The user asks "Explain the code." and the SAS Viya Copilot responds with a detailed explanation of the SAS code. The explanation describes the use of PROC LIFETEST for survival analysis, the variables involved, and the stratification by Treatment_Group. A mouse cursor points to the chat window.

On the right, the "Code Generation.sas" file is open in the editor. The code includes comments and SAS procedures for frequency tables, scatter plots, Kaplan-Meier survival functions, and proportional hazards models.

```
43
44 /* Create a frequency table on occurred */
45 proc freq data=mycas.pharma;
46     tables Event_Occurred;
47 run;
48
49 /* Generate a scatter plot of Age_at_Study_Entry against Event_Occurred */
50 proc sgscatter data=mycas.pharma;
51     title "Scatter Plot: Age vs. Time to Event (Group)";
52     plot Age_at_Study_Entry*Time_to_Event / group=Event_Occurred;
53 run;
54 title;
55
56
57 /* Estimate a Kaplan-Meier Survival Function for Time_to_Event and Event_Occurred by Treatment_Group */
58 proc lifetest data=mycas.pharma;
59     time Time_to_Event*Event_Occurred(0);
60     strata Treatment_Group;
61 run;
62
63
64 /* Estimate a proportional hazard model for Time_to_Event and Event_Occurred against Treatment_Group and Age_at_Study_Entry */
65 proc phreg data=mycas.pharma;
66     model Time_to_Event*Event_Occurred(0)=Treatment_Group Age_at_Study_Entry;
67 run;
68
69
70
71
```

Explain the Code

This SAS code is using the PROC LIFETEST procedure to perform survival analysis on the dataset named "mycas.pharma". The "Time_to_Event" variable is the time variable that represents the duration until the event of interest occurs, and the "Event_Occurred" variable is a binary variable indicating whether the event has occurred (1 for event occurrence, 0 for no event occurrence). This is specified using the "Time_to_Event*Event_Occurred(0)" syntax. The "strata" statement is used to stratify the analysis by the variable "Treatment_Group", which means that the analysis will be performed separately for each unique value of the "Treatment_Group" variable. Overall, this code is performing survival analysis using the Cox proportional hazards model to assess the relationship between the time to event and the treatment group, while taking into account any stratification specified by the "Treatment_Group" variable.

Ln 58, Col 1 (105 selected) Spaces: 2 UTF-8 LF SAS



SAS Code Generation

The screenshot displays the SAS Viya Copilot interface. The main editor shows SAS code for generating a scatter plot and survival analysis. A callout box highlights the `proc phreg` code block, with a mouse cursor pointing to the `mod` statement. A blue box on the right contains the text "Add Comments to the Code".

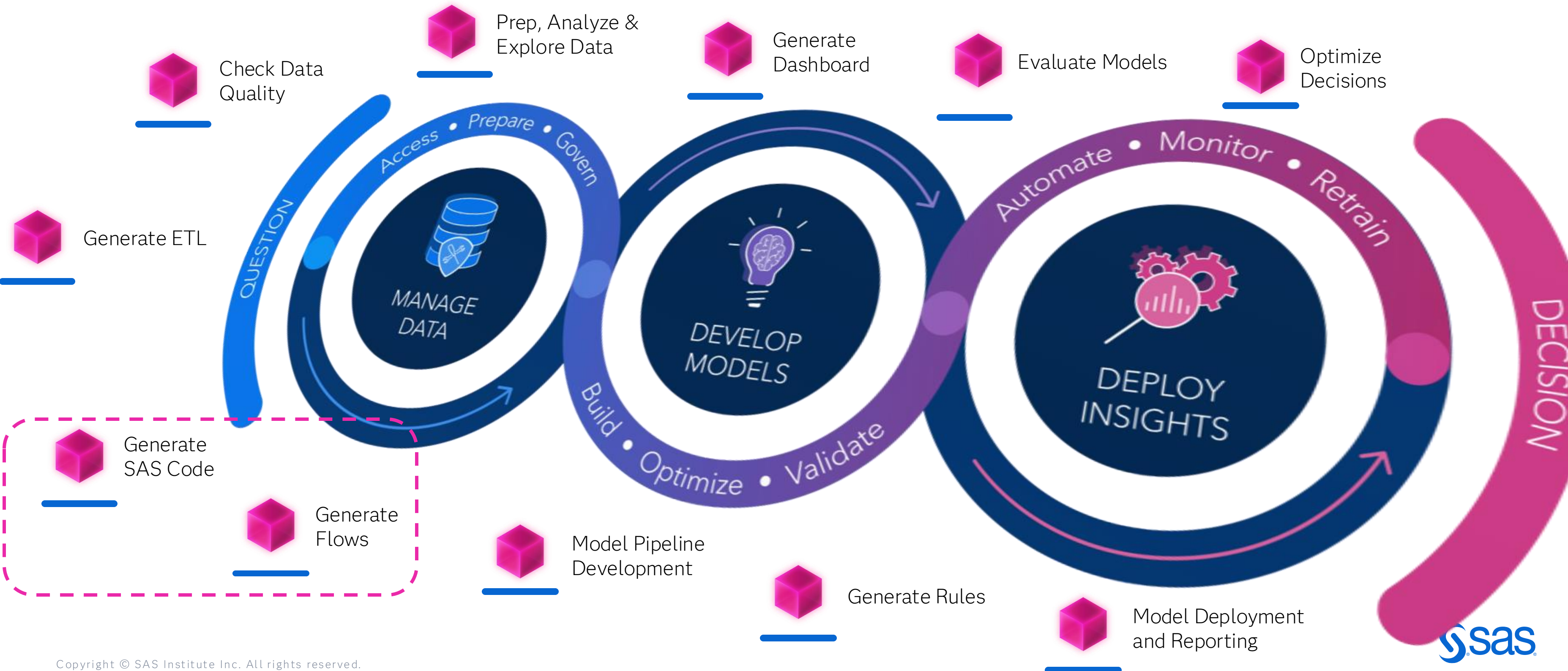
```
50 proc sgscatter data=mycas.pharma;
51   title "Age at Study vs Time to Event Grouped by Event Occurrence";
52   plot Age_at_Study_Entry*Time_to_Event / group=Event_Occurred;
53 run;
54 title;
55
56 /* Estimate a Kaplan-Meier Survival Function for Time_to_Event and Event_Occurred by Treatment_Group */
57 proc lifetest data=mycas.pharma;
58   time Time_to_Event*Event_Occurred(0);
59   strata Treatment_Group;
60 run;
61
62
63
64 /* Estimate a proportional hazard model for Time_to_Event and Event_Occurred against Treatment_Group and Age_at_Study_Entry */
65 proc phreg data=mycas.pharma;
66   mod Time_to_Event*Event_Occurred(0)=Treatment_Group Age_at_Study_Entry;
67 run;
68
69
70
71
```

Add Comments to the Code

Ln 65, Col 1 (111 selected) Spaces: 2 UTF-8 LF SAS

Your AI assistant for the Data and AI Lifecycle

Ease the barrier to data analysis and AI development





SAS Viya Copilot for Data Processing Code Flows

SAS® Studio - Develop Code and Flows

New Options View Open Save All SAS Studio compute context

Start Page demo.flw x demo.sas +

Run [Icons] Add View [Icons]

Flow Submitted Code and Results

AIR Query Sort out

Query Options Node Notes

Columns Select Join Filter Sort Output Options

Expression builder Reset filter

1 t1.'DATE'> -3836

SAS Content: /Public/demo.flw Recover (1) Submission (0)

SAS Viya Copilot

January 1, 1960.

Here's how you can execute this conversion in a SAS environment:

1. Use a DATA step to create a new variable with the DATE9. format.

```
1 /* Step 1: Define the numeric SAS date value */
2 data _null_;
3   numeric_date = -3836; /* The given number of days */
4
5 /* Step 2: Convert the numeric date to a SAS date in `DATE9.` format */
6   formatted_date = put(numeric_date, DATE9.);
7
8 /* Step 3: Display the formatted date */
9   put formatted_date=;
10 run;
```

The code above assigns the DATE9. format to -3836, which should convert it to the SAS date approximately around

Ask SAS Viya Copilot...

AI-generated content. Evaluate accuracy and suitability before use.



Benefits of SAS Viya Copilot

Summary Report Details Errors Error Detail Priority Selector

Procs with Info-Packs

GitHub Copilot

17.3% mean score

Our model beats Github Copilot by 49.8%

Results taken over 93 procs

Result Paths

Our Model

67.1% mean score

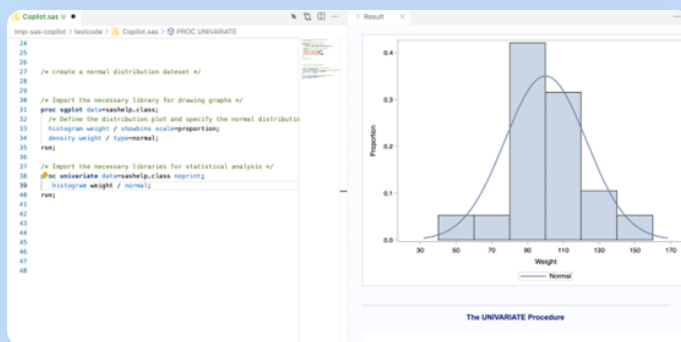
proc	GitHub Copilot	Phi-3-mini2025.6.0		
anova	0	0		
assess	0	0.736		
assessbias	0	0.928		
astore	0.125	0.861		
bchoice	0	0.938		
binning	0	0.6		
bnet	0	0.658		
cancorr	0	1		
cardinality	0.288	0.915		
catmod	0	0		
cattrtransform	0	0.54		
correlation	0	0.593		
cspade	0	1		
datastep	0.79	0.822		
dynbnet	0	0.837		
export	0.978	0.889		
factmac	0	0.947		
fastclus	0.2	0.853		
fastknn	0	0.701		
fism	0	0.667		
fittedqnet	0	0		
flattenimages	0	0.88		
forest	0	0.914		
freq	0.753	0.982		
gee	0.2	0.6		
genmod	0.2	0.775		
glimmix	0.217	0.535		
glm	0.58	0.75		
glmmod	0	0.8		
glmselect	0.775	0.114		

SAS and Generative AI (GenAI)



■ SAS Viya Copilot to accelerate user productivity

- Conversationally query SAS Viya to increase user productivity, further democratizing analytics

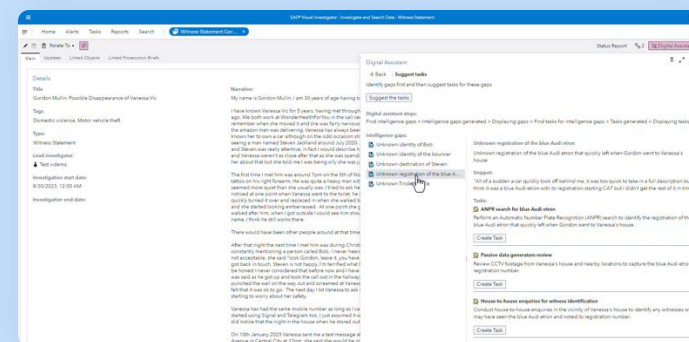


example: SAS Code Generation



■ Specialized SAS Viya Copilots to accelerate industry workflows and processes

- GenAI driven assistants to accelerate domain-specific workflows

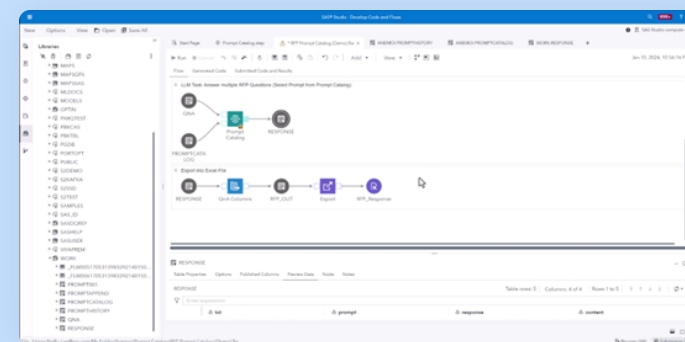


example. Clinical Data Lifecycle



■ SAS Viya to help build your LLM-based applications to unlock value in your proprietary business assets

- Flexible SAS Viya GenAI building blocks to help with development and orchestration of your LLM application stack



example. Build Prompt Library using existing SAS Viya APIs and SAS Studio steps



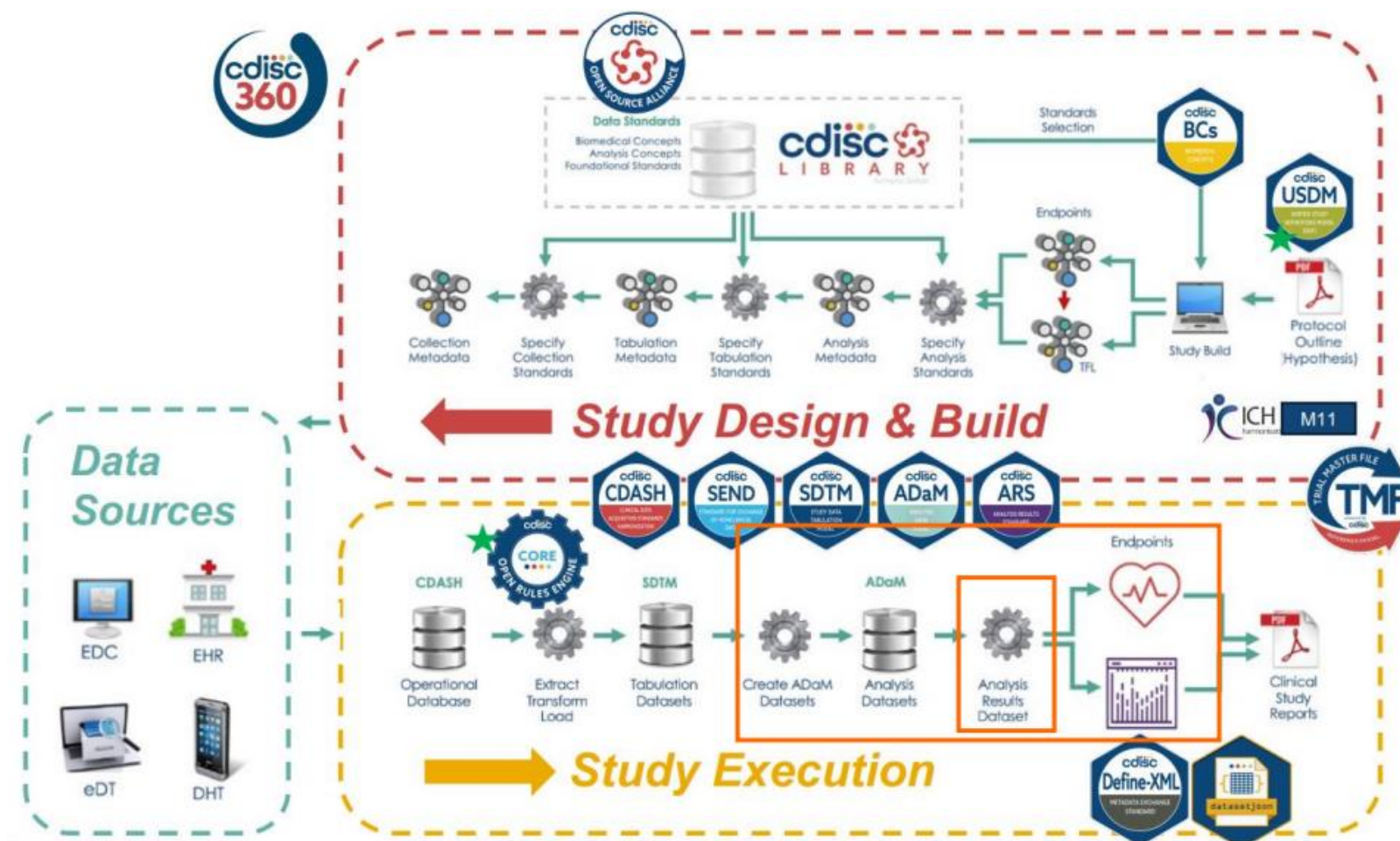
■ Standalone GenAI services to enable your toolchains, like a synthetic data generation service

- API-first low-code/no-code synthetic data generation and validation experience



example. Synthetic data generation API

Automation of the Drug Submission Process using a Multi agent framework



Raw data to SDTM format

SDTM Validation

SDTM to ADaM

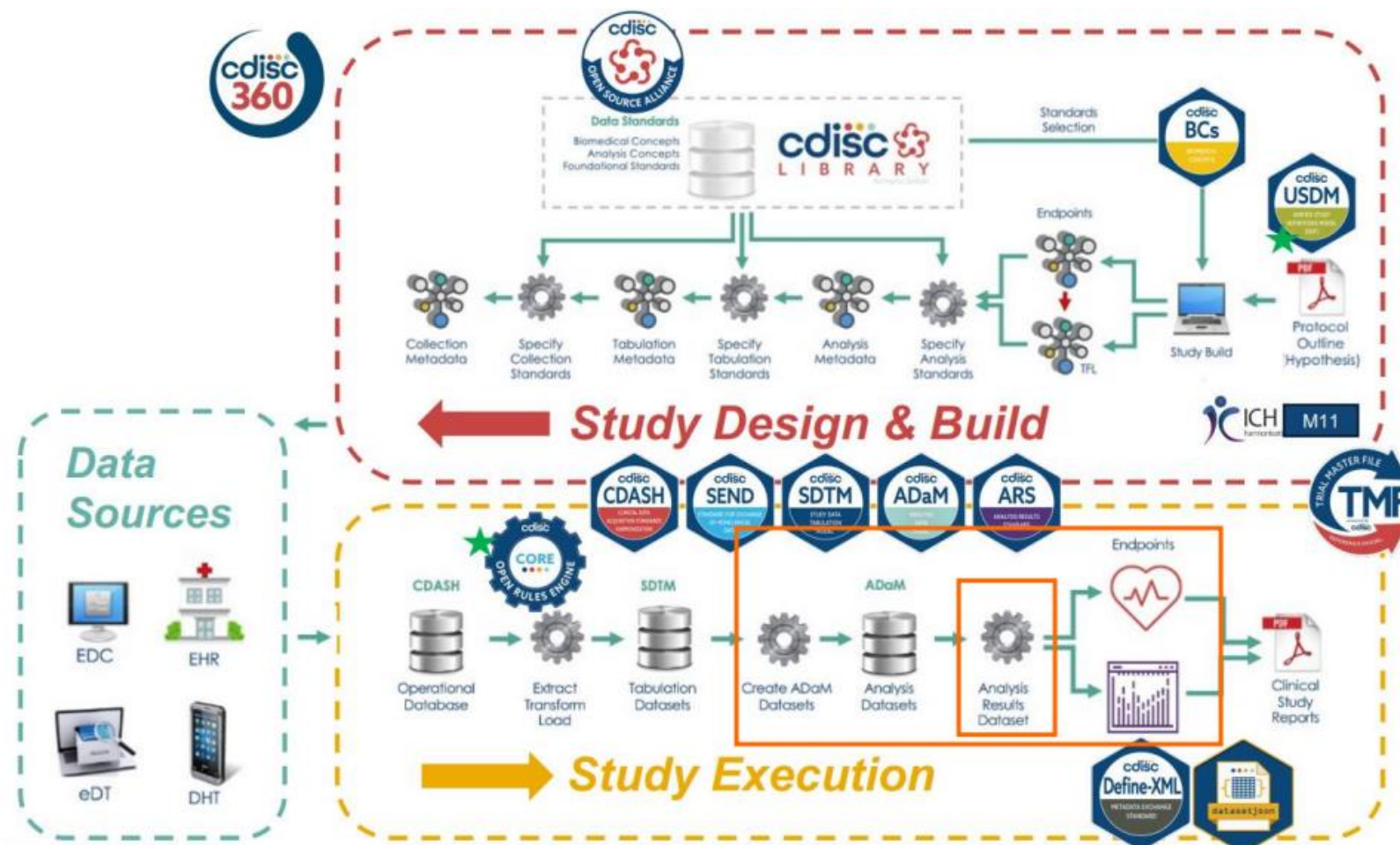
ADaM Validation ★

Creation of define.xml

Creation of TLFs



Automation of the Drug Submission Process using a Multi agent framework



Raw data to SDTM format

SDTM Validation

SDTM to ADaM

ADaM Validation

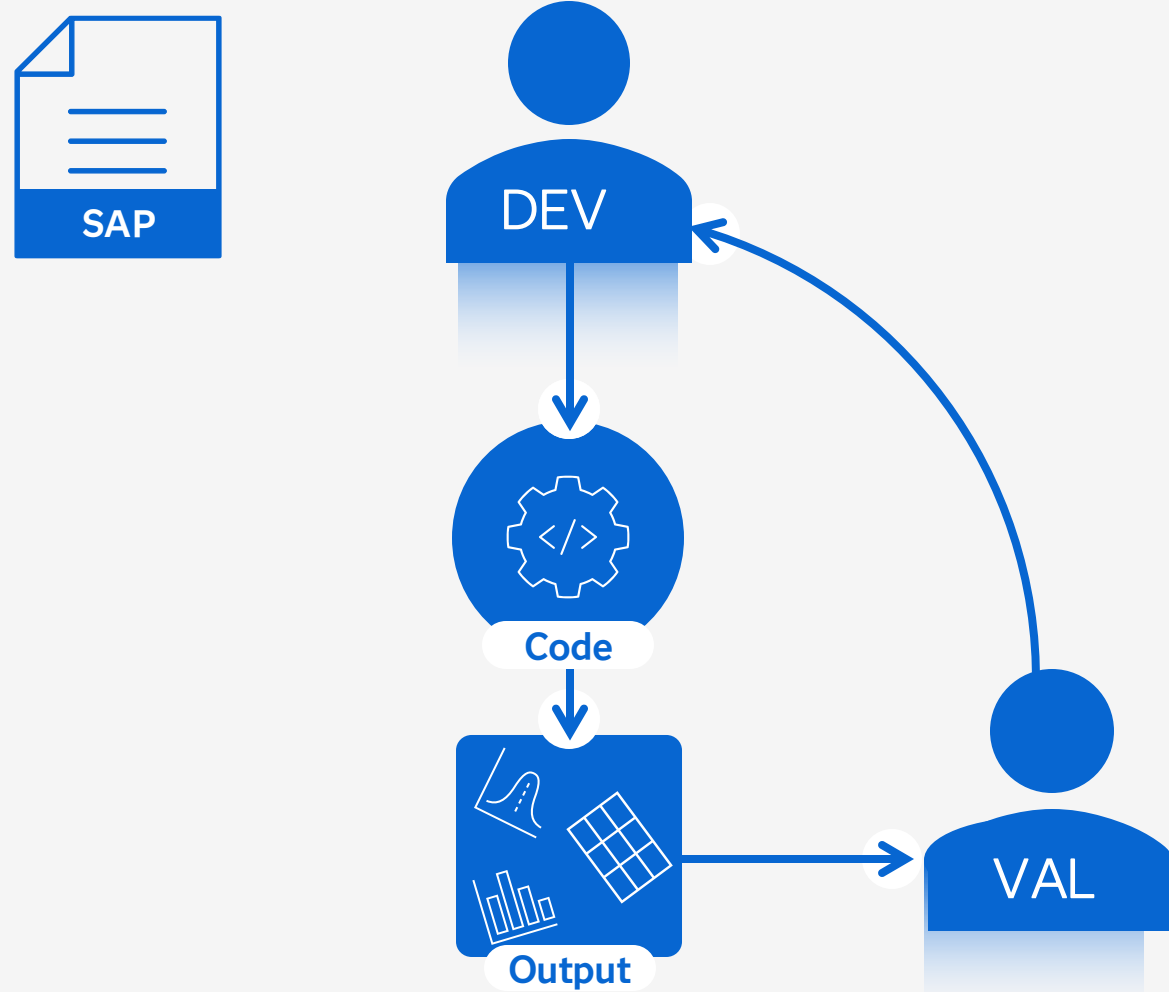


Creation of define.xml

Creation of TLFs



Validation of Statistical Output

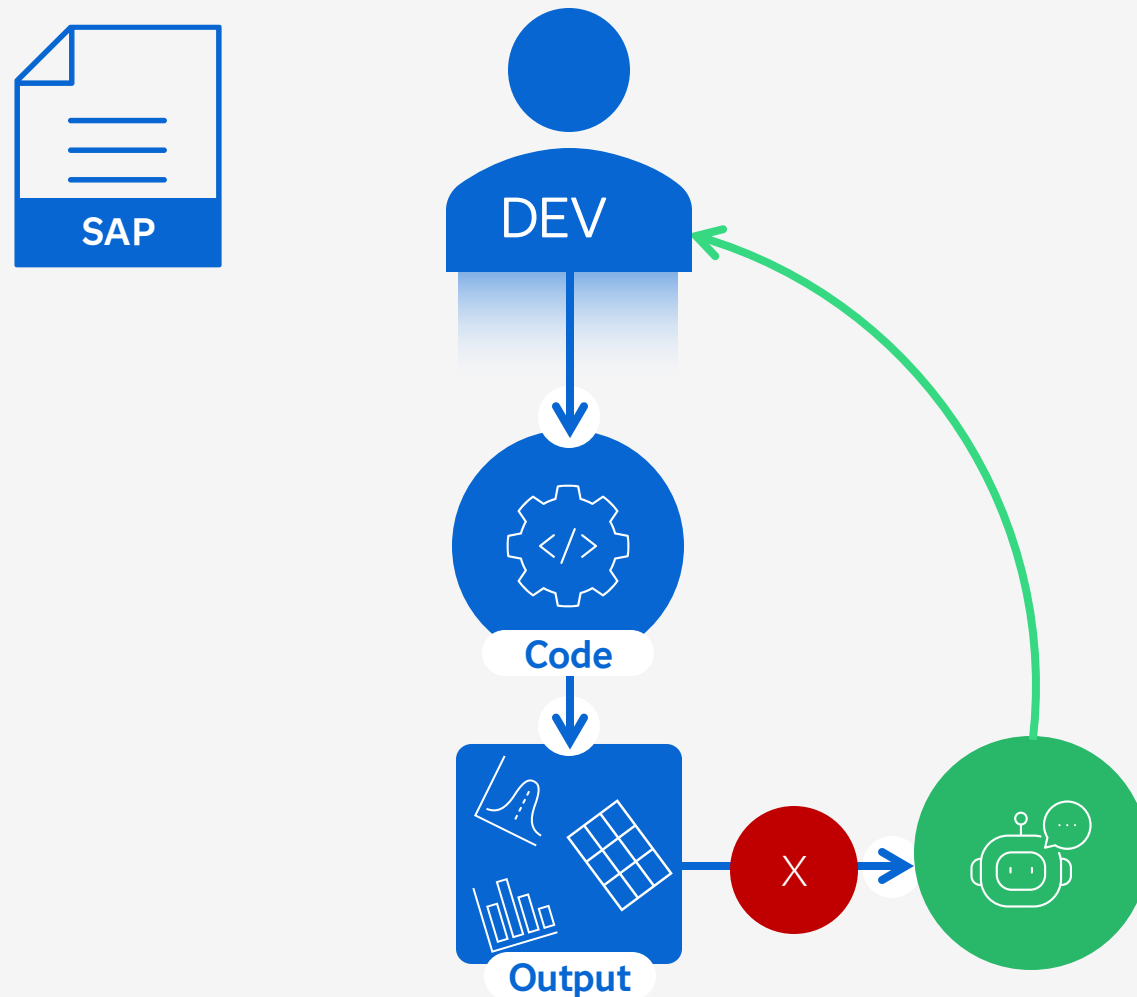


VALIDATION OF RESULTS

TIME CONSUMING



AI Validation Challenges



LEGAL QUESTIONS

- Ethics
- Confidentiality

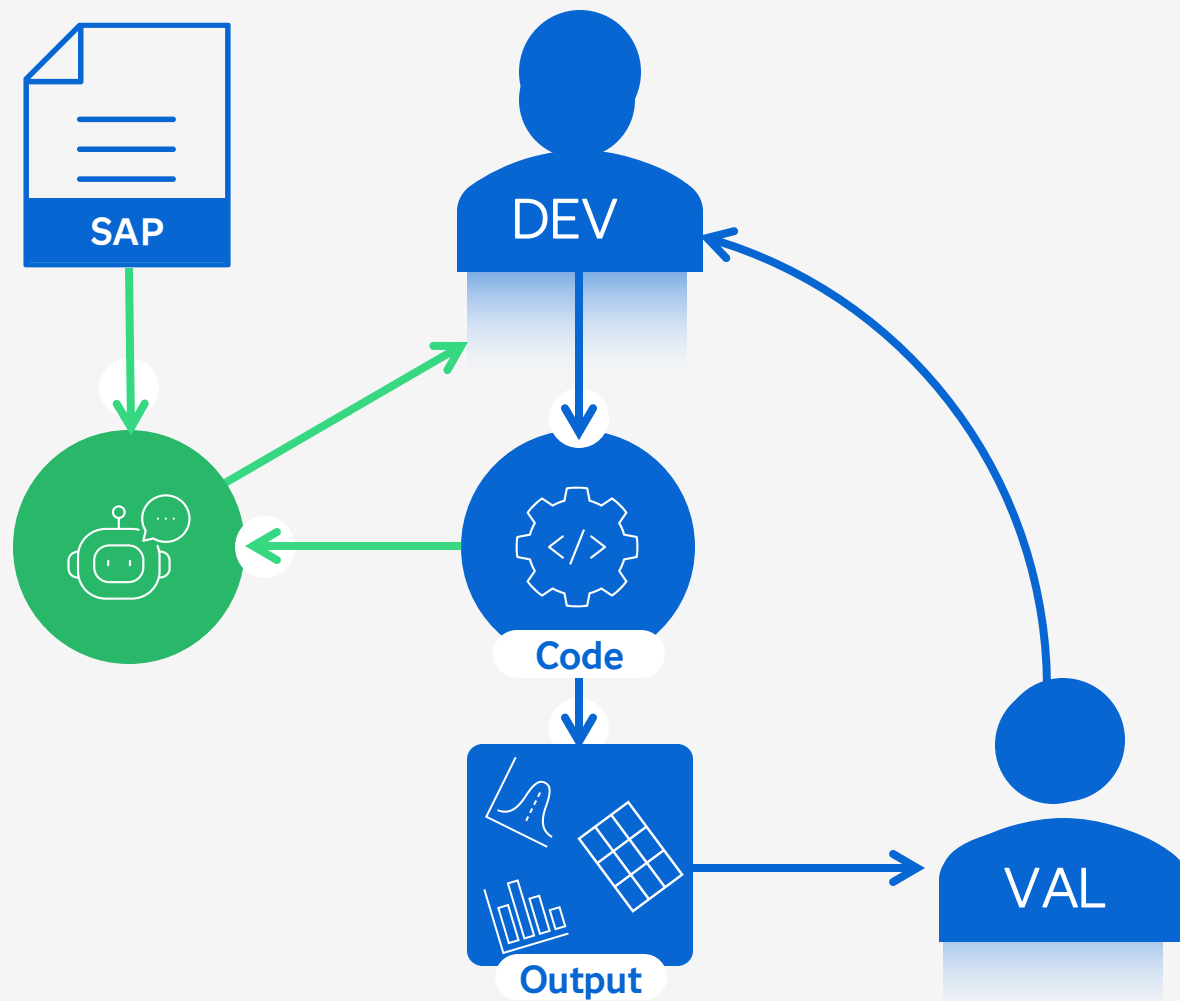
TECHNICAL ISSUES

- Hallucinations
- Version control

FINANCIAL IMPACT

- Data limits
- Token usage

Validation of Statistical Output



INPUT

- SAP + Code
- Confidentiality

INSTRUCTIONS

- By section
- Incremental

Select a task to perform on the ADSL program (adsl.sas):

- ☐ (None)
- ☐ Validate Presentation of Treatments
- ☐ Validate Rounding of Derived Variables
- ☐ Validate Analysis Sets
- ☒ Mark Validation as Finished

Select a task to perform on the ADaPer program (adaper.sas):

- ☐ (None)
- ☐ Validate Consistency Between adaper.sas and adsl.sas
- ☒ Mark Validation as Finished

Select a task to perform on the 24_adv.sas:

- ☒ (None)
- ☐ Normal Range
- ☐ Include All Parameters?
- ☐ Check ANL01FL
- ☐ Mark Validation as Finished

ADaM Code Validation using LLM

Step 1: Get SAP.

Drag and drop the SAP here, or click to browse



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

Browse files

No SAP file uploaded yet.

Step 2: Get SAS Code.

Step 2.1 : Please enter the folder path containing the SAS code files for ADaM data creation.

\STAT\Analyses\Programs\AI\02_testrun02\repository\1.ADAI

Example:

\STAT\Analyses\Programs\AI\02_testrun02\repository\1.ADAM\

Step 2.2: Please select the SAS files that you would like to check from the .sas files found in
/STAT/Analyses/Programs/AI/02_testrun02/repository/1.ADAM:

☒ Select All SAS files

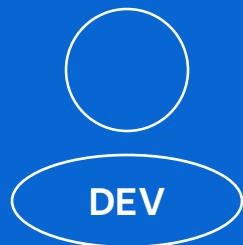
Selected SAS files:

▼ [
0 : "01_adsl.sas"
1 : "02_adaper.sas"
2 : "24_adv.sas"

Example: Validate Normal Range in advs.sas

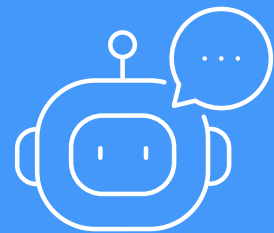
Results

Original

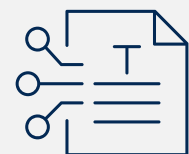


```
else if kupcase(vstestcd) eq 'DIABP' then do;  
    ATOXGRHN = 0 + (aval ge 141) + (aval ge 151) + (aval gt 155);  
    if not missing(atoxgrhn) then ATOXDSCH = 'HYPERTENSION (DIASTOLIC)';  
end;
```

LLM
Update



```
else if kupcase(vstestcd) eq 'DIABP' then do;  
    /* Corrected the ranges for Hypertension (Diastolic Blood Pressure) */  
    ATOXGRHN = 0 + (aval ge 91) + (aval ge 96) + (aval gt 100);  
    if not missing(atoxgrhn) then ATOXDSCH = 'HYPERTENSION (DIASTOLIC)';  
end;
```



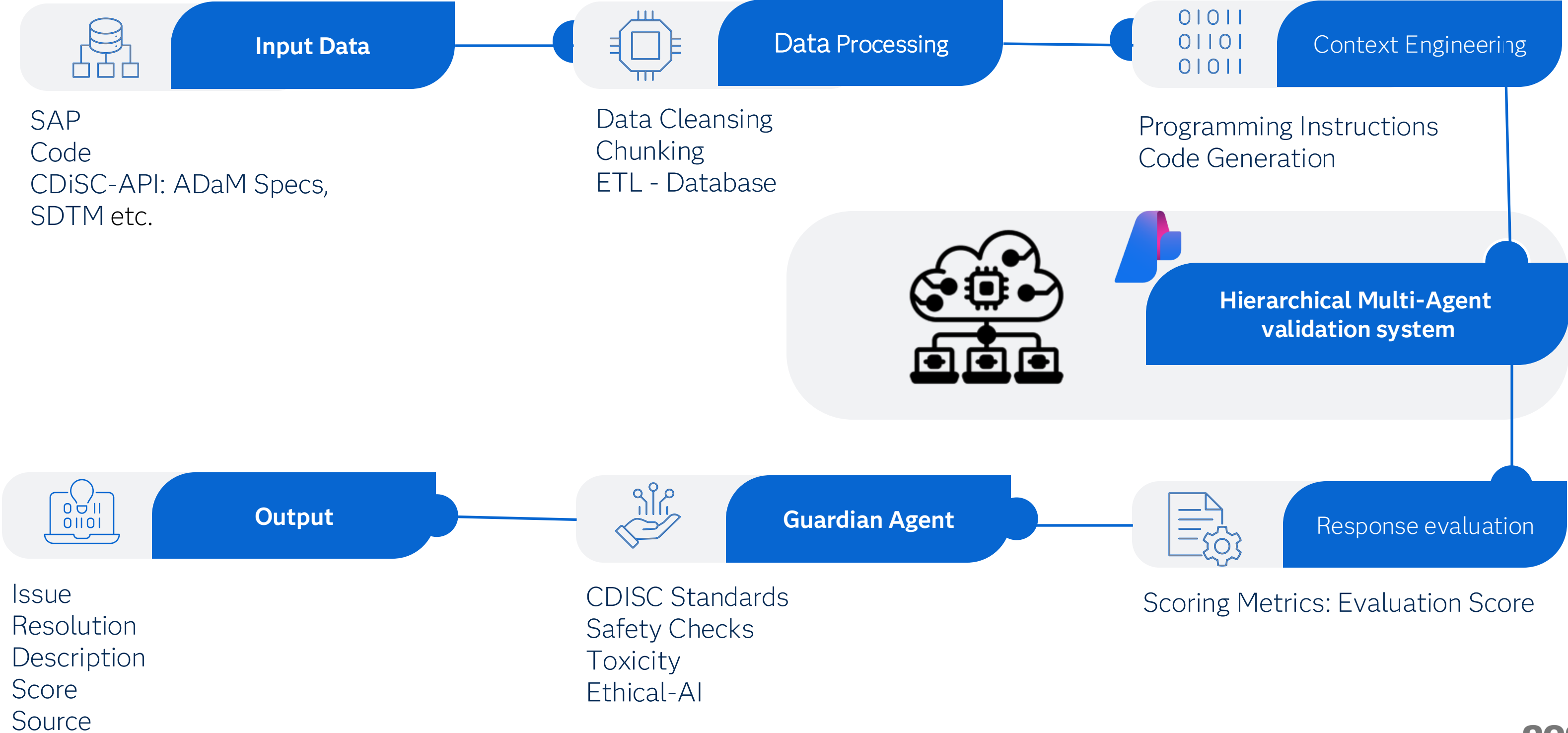
AI annotates any changes



Code structure is kept

Agentic Workflow for ADaM Validation

Leveraging Multi-Agent system to Ensure Data Integrity and Compliance



Clinical Trials Workflow Vision

**Clinical Trial
Protocol**

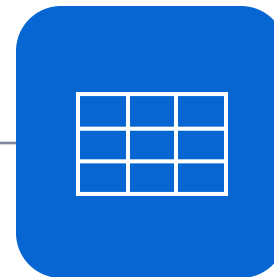


**Data
Collection**



eCRF / eSource

**Data
Tabulation**



SDTM

**Statistical
Analysis**



ADaM / TLFs

**Clinical Study
Report**



Raw data to SDTM format

SDTM to ADaM

ADaM Validation

Creation of define.xml

Creation of TLFs

Thank you !



Soundarya Palanisamy
Soundarya.Palanisamy@sas.com



**The Global Healthcare
Data Science Community**