

ML31

Batch Cleaning of SAS Programs for Submission Readiness:

An AI and Python Hybrid Approach

Qisen Luan, Qiong Wei, Lixin Gao

VeraMed

Agenda

01 Overview and Motivation

End-to-End Workflow for SAS Programs Cleanup

- Comment Extraction
- Comment Classification
- Batch Cleanup
- Validation and QC

02

03 Case Study and Performance Assessment

04 Summary



Overview and Motivation

Challenges in Preparing SAS Programs for Submission

- Submission-ready SAS programs must be clear, concise, and free of irrelevant content.
- Over time, SAS programs accumulate large volumes of mixed-use comments (documentation vs. noise):
 - Narrative comments (useful vs. useless)
 - Debug notes (generally useless)
 - Comment-out codes (generally useless)
- Manual cleanup is:
 - Time-consuming
 - Error-prone
 - Inconsistent across programmers
 - Lack of traceability and hard to audit

4.1.2.10 Software Programs

Sponsors should provide the source code used to create all ADaM datasets, tables, and figures associated with primary and secondary efficacy analyses. Sponsors should submit source code in single byte ASCII text format. Files with MS Windows executable extensions (.cmd, .com, and .exe) should NOT be submitted. For a list of acceptable file extensions, refer to the document entitled *Specifications for File Format Types Using eCTD Specifications*.³²

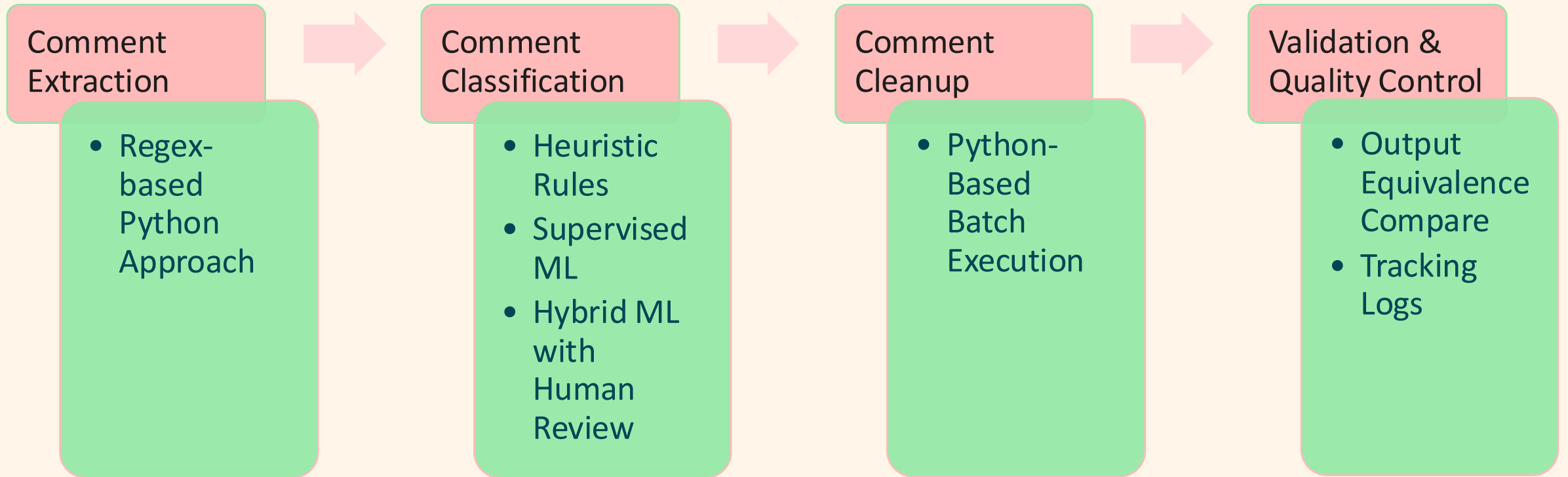
[Study Data Technical Conformance Guide - Technical Specifications Document | FDA](#)





End-to-End Workflow

End-to-End Workflow



Step 1. Regex-Based Comment Extraction

Comment type	Regex pattern
Block /* ... */	<code>r'/*. *?\' with DOTALL</code>
Starblock (* at line start)	<code>r'^\s*\s*\'</code>
Inline (*...; or %*...;)	<code>r'(?i)(?: (?<=;) ^)\s*%?\' \s* [^;]*;\'</code>
Comment keyword (comment ...;)	<code>r'(?i)(?: (?<=;) ^)\s*comment\b (?!\s*=) \s* [^;]*;\'</code>

- Support all typical SAS comment types
- Preserve positional metadata
- Output a structured CSV file for downstream review

A		B
1	Comment Type	Extracted Comments
		/* ods listing close; options orientation=landscape nodate nonumber center nobyline missing=""; ods escapechar='`'; ods rtf file="&_xpathroot\prod\patientprofile\z-test-rpt.rtf" notoc_data style=style_land_8_c startpage=NO;  ods rtf startpage=NOW; %ppheads; ods rtf text="` S={font_size=8pt font_weight=bold just=l}AE (Part 1)"; %rptae; ods rtf close; */
2	block	
3	block	/*%ppblktextline(nblkline=4); */
		***** **; * call to process with two measurements of bottom margin *****
4	starblock	**;
5	inline	*** NOTE: Temporarily setting PPFLAG to null for all records ***;
6	inline	* define gap / style(column)=[cellwidth=0.2%] " ";
7	inline	*if subject='1103-003' then dsdat='nnzzzyyyy';
8	inline	%*** define pagesize and linesize macros ***;
9	inline	*ods rtf text="` S={font_size=8pt just=l}";
10	inline	*Get raw data needed and set together;
11	inline	*title2 j=l "Study: xxxx";
12	inline	*Strip out special characters;
13	inline	*line @1 ' ';
14	inline	*%uline;

Step 2. Comment Classification



Approach A: Heuristics Rules (Simple Rule Set: Keep Header Only)

- Pros: easy, fast, deterministic
- Cons: remove valuable narrative comments.

Rule ID	Condition (evaluated in order)	Decision	Typical example
R1	Program header block (contains all required header keywords such as author name, programming date/purpose, etc.)	Keep	Standard header block
R2	R1 Excluded	Remove	All other comments

Step 2. Comment Classification

	A	B	C
1	Comment Type	Extracted Comments	Classification
		/* ods listing close; options orientation=landscape nodate nonumber center nobyline missing=""; ods escapechar='`'; ods rtf file="&_xpathroot\prod\patientprofile\z-test-rpt.rtf" notoc_data style=style_land_8_c startpage=NO;  ods rtf startpage=NOW; %ppheads; ods rtf text="`S={font_size=8pt font_weight=bold just=l}AE (Part 1)"; %rptae; ods rtf close; */	
2	block	*/	Remove
3	block	/*%ppblktextline(nblkline=4); */	Remove
4	starblock	***** **; * call to process with two measurements of bottom margin ***** **;	Remove
5	inline	*** NOTE: Temporarily setting PPFLAG to null for all records ***;	Remove
6	inline	* define gap / style(column)=[cellwidth=0.2%] " ";	Remove
7	inline	*if subject='1103-003' then dsdat='nnzzzyyyy';	Remove
8	inline	%*** define pagesize and linesize macros ***;	Remove
9	inline	*ods rtf text="`S={font_size=8pt just=l}";	Remove
10	inline	*Get raw data needed and set together;	Remove
11	inline	*title2 j=l "Study: xxxx";	Remove
12	inline	*Strip out special characters;	Remove
13	inline	*line @1 ' ';	Remove
14	inline	;%uline;	Remove

Step 2. Comment Classification



Approach B: Heuristics Rules (Complex Rule Sets)

- Pros: transparent, add flexibility to keep some useful comments
- Cons: rule maintenance grows with codebase diversity, human review effort grow with comments amount and complexity.

Rule ID	Condition (evaluated in order)	Decision	Typical example
R1	Program header block (contains all required header keywords)	Keep	Standard header block
R2	Decoration-only separators (all non-empty lines are only *, -, =)	Keep	*****, ----, =====
R3	Macro invocation statement (including multi-line %mymacro(...);)	Remove	%predata(in=dm); or multi-line %mpgbreak2(...);
R4	SAS structural keywords present, e.g. “proc xxx”, “data”, “run;”, “quit;”, “adam.[dataset name]”, “set [dataset name]”, “if ... then ... ;” that appears as DATA-step conditional statement, macro-language constructs, etc.	Remove	if cmongo1=1 then pedate='Ongoing'; work.adsl, x=y; set adsl; %macro xxx; %if ... %then ...; %let ...
R5	No SAS “code indicators” and no SAS keyword match	Review	Pure narrative text

Step 2. Comment Classification

	A	B	C
1	Comment Type	Extracted Comments	Classification
2	block	<pre>/* ods listing close; options orientation=landscape nodate nonumber center nobyline missing=""; ods escapechar='`'; ods rtf file="&_xpathroot\prod\patientprofile\z-test-rpt.rtf" notoc_data style=style_land_8_c startpage=NO;  ods rtf startpage=NOW; %ppheads; ods rtf text="` S={font_size=8pt font_weight=bold just=l}AE (Part 1)"; %rptae; ods rtf close; */</pre>	Remove
3	block	<pre>/*%ppblktextline(nblkline=4); */</pre>	Remove
4	starblock	<pre>***** **; * call to process with two measurements of bottom margin ***** **;</pre>	Keep
5	inline	<pre>*** NOTE: Temporarily setting PPFLAG to null for all records ***;</pre>	Review
6	inline	<pre>* define gap / style(column)=[cellwidth=0.2%] " ";</pre>	Remove
7	inline	<pre>*if subject='1103-003' then dsdat='nnzzzyyyy';</pre>	Remove
8	inline	<pre>%*** define pagesize and linesize macros ***;</pre>	Review
9	inline	<pre>*ods rtf text="` S={font_size=8pt just=l}";</pre>	Remove
10	inline	<pre>*Get raw data needed and set together;</pre>	Review
11	inline	<pre>*title2 j=l "Study: xxxx";</pre>	Remove
12	inline	<pre>*Strip out special characters;</pre>	Review
13	inline	<pre>*line @1 ' ';</pre>	Remove
14	inline	<pre>*%uline;</pre>	Remove

Step 2. Comment Classification



Approach C: Supervised ML Prediction

- Train classifier on labeled comment snippets
 - Input: labeled comment CSV (comment, keep?)
 - Preprocess: strip SAS delimiters (`/* */`, leading `* / %*`, trailing `;`), normalize whitespace
 - Vectorize: TF-IDF char 3–5 grams + word 1–2 grams (optionally add sentence embeddings)
 - Model: Logistic Regression (`class_weight=balanced`) or Linear SVM
 - Calibrate: convert scores $\rightarrow$ $P(\text{useful})$ (Platt / isotonic)
- Classify a large set of comments using the trained classifier, calculate $P(\text{useful})$ and output predicted decision based on pre-specified thresholding:
 - $P(\text{useful}) \geq 1 - \text{threshold}$: keep
 - $P(\text{useful}) \leq \text{threshold}$: remove
 - `review_flag = (threshold <  $P(\text{useful})$  < 1 - threshold)`

Step 2. Comment Classification

Approach D: Supervised ML Prediction + Human-in-Loop Review

- Classification accuracy depends on many factors; goal is to iteratively train the classifier, increase prediction accuracy and decrease human review effort.

	A	B	C	D	E	F	G	H	I	J
1	file	type	startline	comment	label	reason	keep?	useful_pre	useful_pre	Correct?
1227	C:\Users\c\block		20	/* proc print noobs width=min; var form analytename labunits stdunits stdvalue analytevalue cf; run; */	code	103	N	0.10712	not_usefu	Y
1228	C:\Users\c\inline		315	***outlier QTCF out01fl >500 & baseline<=500 out02fl >480 & baseline<=480 out03fl >450 & baseline<=450***;	code	0	Y	0.80083	useful	Y
1229	C:\Users\c\block		1	/* purpose: for all 4.x tables (Study Drug Exposure) using ADEXSUM date: 2020-05-17 created by: Tom Zhou */	keep	2	Y	0.78711	useful	Y
1230	C:\Users\c\block		83	/*if SHTLBL1='Total Bilirubin (High)' then do; byvarn=5; byvar='Total bilirubin (HIGH)'; output; end;*/	code	102	N	0.05338	not_usefu	Y
1231	C:\Users\c\starblock		124	***** * Combine SDTM LB and SUPPLB DATA *****;	keep	2	Y	0.46671	not_usefu	N
1232	C:\Users\c\starblock		33	***** * Combine SDTM QS and SUPPQS DATA *****;	keep	2	Y	0.46671	not_usefu	N
1233	C:\Users\c\block		1051	/*1/20/2022, change from 2002, since 'CR' 'PR' with A6WKFL^='Y' already removed*/ /*else CAVALC='PD';*/	code	0	Y	0.63576	useful	Y
1234	C:\Users\c\block		178	/*if prstdtc ne "" and epoch = "" then do;*/ /*if prstdtc<rfstdtc then epoch = 'SCREENING';*/ /*end;*/	code	102	N	0.04235	not_usefu	Y
1235	C:\Users\c\block		638	/*if avalc="PR" then fin=" ";*/ /* if avalc="SD" then c=0;*/ /*if index(visit,"UNSCHEDULE")=0 then do;*/	code	102	N	0.05311	not_usefu	Y

Step 3. Python-Based Batch Comment Cleanup



File Rewriting Process

- Read in comments classification CSV file, and rewrite each .sas file to a new folder
 - preserving essential comments (marked as “Keep” in the classification CSV file)
 - removing useless comments (marked as “Remove” in the classification CSV file)

Tracking Log Generation

- Generate a detailed tracking log which records file paths, removed line ranges, classification reasons, and timestamps for auditability.

Preservation of Originals

- Original files remain untouched, ensuring changes are reversible and traceable for compliance.

Step 4. Validation and Quality Control



Dataset Comparison

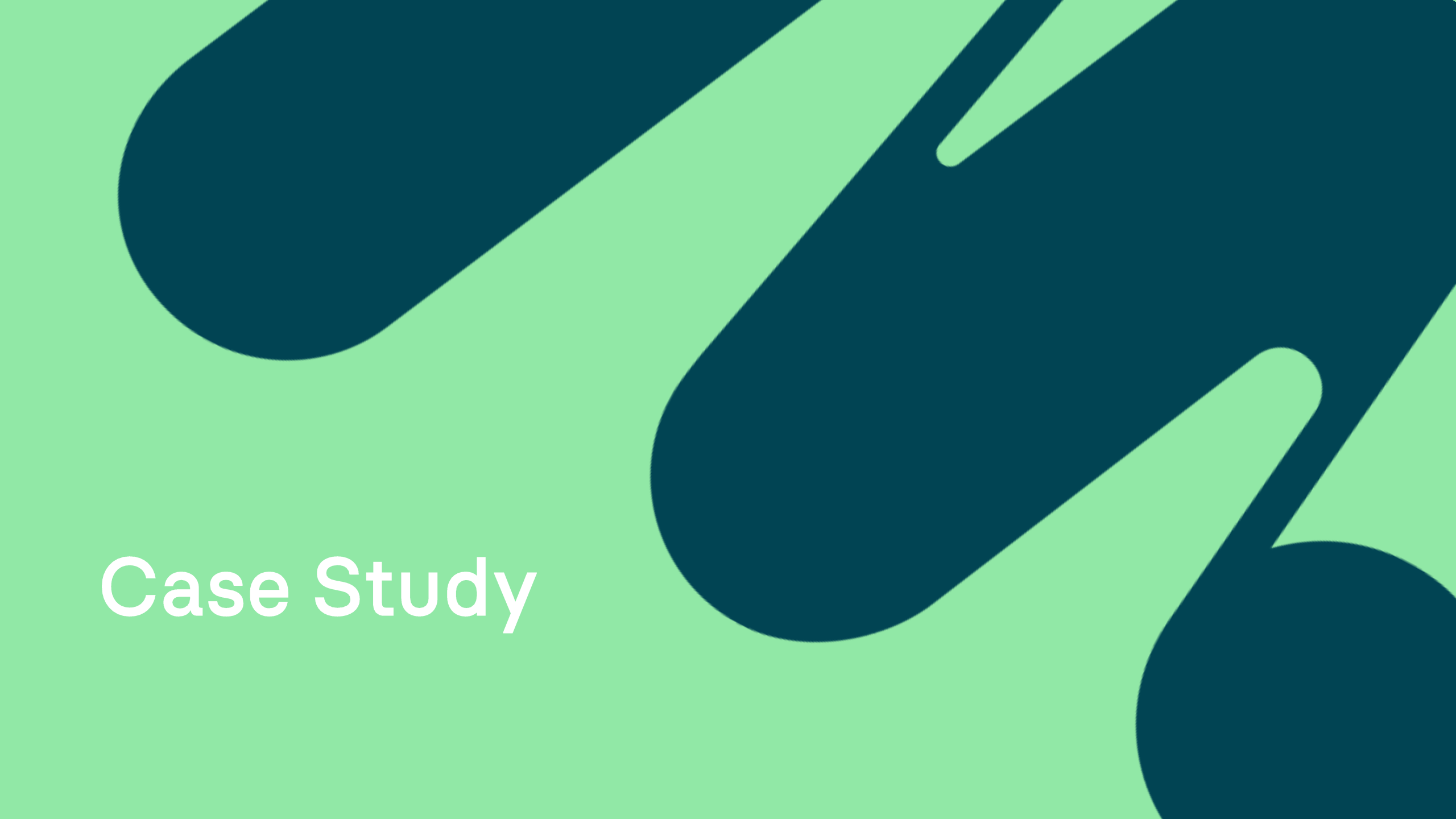
- SDTM/ADaM Datasets: PROC COMPARE with strict criteria

Report File Comparison

- RTF: RTF compare tools
- PDF: Adobe Acrobat Pro

Quality Control Outcomes

- Goal: 100% match between original datasets/outputs and those produced by cleaned programs.



Case Study

Case Study

1. Input Codebase

- Programs: ~160
 - ADaM: ~15
 - TFL: ~52
 - Macro: ~82
- Extracted Comments: 1118
 - Block comment: 638
 - Starblock comment: 251
 - Inline comment: 229

Example Program Before Cleaning

```
dm 'log;clear;out;clear;';
/*****
Protocol Number: XXX
Program Name: q-admh.sas
Program Purpose: qc ADMH
Date Created: 2025-XX-XX
Programmer: ABC XYZ etc.
*****/
%include "..\..\macros\setup.sas";
data qc1;
    length usubjid $40;
    merge sdtm.mh(in=a) adam.adsl(keep=&adslvars);
        by usubjid;
        if a;

run;
/*This is a valid comment in SAS*/
data final;
    set qc1;

/*
    array oo[2] mhstdtc mhendtc;
    array ll[2] asdtcl aendtc;
    do i=1 to 2;
        if length(oo[i])=7
            then ll[i]='--' || substr(put(input(catx('-',oo[i],'01'),yymmdd10.),date9.),3);
    end;
*/
/*%somemacro;%/
Run;
proc compare data=admh&keep comp=qc&keep listall method=absolute
criterion=0.000001;
    id &id;
run;
```

Case Study

2. Output Deliverables and Performance Assessment

Approach	Description	Comment Classification Accuracy	Human Review Effort
A	Keep programs header, remove others	100%	Minimal
B	Heuristic rules	98% for long comments	Moderate
C	ML (TF-IDF/Emb + Logistic/SVM)	82%	Low to Moderate
D	Hybrid (Rules→ML→Review)	100%	Low

Example Program After Cleaning

```
dm 'log;clear;out;clear;';
/*****
Protocol Number: XXX
Program Name: q-admh.sas
Program Purpose: qc ADMH
Date Created: 2025-XX-XX
Programmer: ABC XYZ etc.
*****/
%include "..\..\macros\setup.sas";
data qc1;
    length usubjid $40;
    merge sdtm.mh(in=a) adam.adsl(keep=&adslvars);
        by usubjid;
        if a;

run;
/*This is a valid comment in SAS*/
data final;
    set qc1;

run;
proc compare data=admh&keep comp=qc&keep listall
method=absolute criterion=0.000001;
    id &id;
run;
```

Summary

1. Fully automated cleanup without oversight risks deleting valuable documentation.
2. Manual-only cleanup is time-consuming, lack of scalability and traceability
3. Human-in-Loop Hybrid Workflow:
 - scalability: scale to large codebases
 - auditability: clear documentation of cleaning logs
 - accuracy: human-in-loop review to ensure 100% output equivalence.

Contact

Name: Qisen Luan, Qiong Wei, Lixin Gao
Organization: BioPier LLC, a Veramed Company
Address: 80 Blanchard Rd Ste 104
City, State ZIP: Burlington, MA 01803
Work Phone: (781) 354-1924
E-mail: qluan@biopier.com
Web: <https://biopier.com/>



Veramed