Expediting Implementation of End-to-end Programming Automations

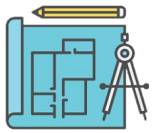
AutoCheng Clinical Data Services LLC

Chengxin(Cheng) Li

July 2022

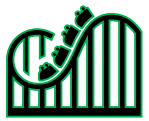
Agenda

Introduction



autoSDTM Driven by SDTMIG

autoADaM Driven by SAP



autoTLF Driven by ARM

Questions

Agenda

Introduction

autoSDTM Driven by SDTMIG

autoADaM Driven by SAP

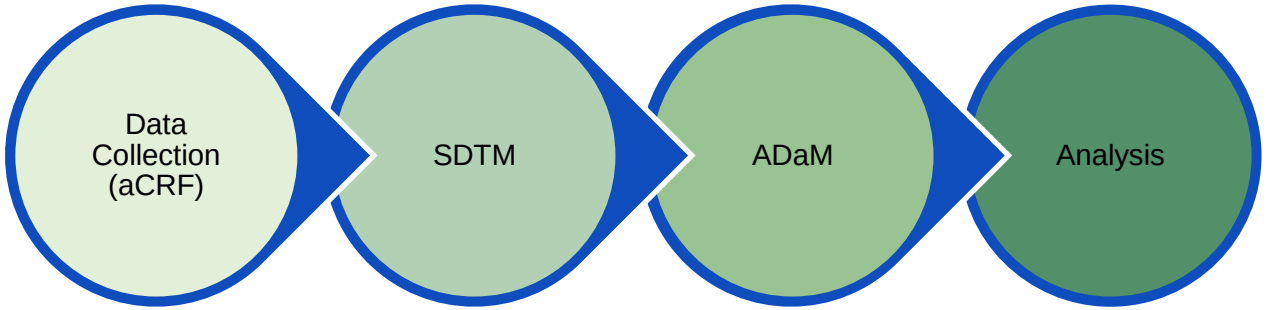
autoTLF Driven by ARM

Questions

Introduction

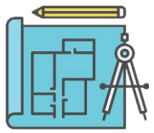
CDISC has been widely deployed, targeting end-to-end streamlined data processing from data collection to analysis in order to ensure traceability, transparency, ease of review, and high quality and efficiency

CDISC programming has become routine work. To prepare for the post-pandemic world, this presentation proposes and summarizes the implementation of end-to-end programming automations under CDISC umbrella, covering autoSDTM (incl. autoACRF), autoADaM, and autoTLF



Agenda

Introduction



autoSDTM Driven by SDTMIG

autoADaM Driven by SAP

autoTLF Driven by ARM

Questions

autoSDTM Driven by SDTMIG

Varied to Rigid

CRF design originated from study protocol and collected from multiple sources (EDC, LAB, eDiary/ePRO, Adjudication, …)

Database metadata setup for the collected data also varied, even implemented with CDASH or CDASH alike

MAPPING

However, the targeted SDTM domains have very rigid metadata defined in the SDTMIG

SDTM mapping required to fully reflect the collected data with very limited derivations, or transformations from horizontal to vertical structure

Imputation NOT allowed

autoSDTM Driven by SDTMIG

Solution

- **Programming mapping solution** no dependence on data management
- **Universal solution** supporting any clinical data formats from unfavorable legacy study to preferable CDASH study
- **Implemented with SAS macros, one macro one domain**
- **Macro parameters aligned with SDTMIG metadata**
- **Best practice in clinical data process incorporated**
- **Mapping specs and plain SAS codes dynamically generated** based on the macro call instance
- **Outputs with submission readiness**
- **Annotation specs dynamically generated** one step away to autoACRF

autoSDTM Driven by SDTMIG

Example of %DM

Parameter	Definition/Input Format	Example
RAWLIB	req., RAW DATA libref name	%str(RAWDATA)
SDTMLIB	req., SDTM output libref name	%str(SDTM)
MISCLIB	opt., MISC libref name storing study misc dataset	%str(MISC)
TOPICDS	req., format: DSNM,raw dataset containing TOPIC variable	%str(DEMOG), or DM
SUBJID	req., DSNM.SRCVAR # DSNM.PRESBJVAR/WHERE, DSNM.PRESBJVAR/WHERE for case of multi-SCRN/multi-ENRL, DSNM optional	%str(PTNO)
DMVARS	req., SDTMVAR=DSNM.SRCVAR, list of direct mapping variables incl domain specific IDs and domain specific vars STUDYID,SITEID SEX,INVID,INVNAM,RACE, COUNTRY, ETHNIC, AGE, AGEU	%str(STUDYID=STUDY # SITEID=SITEID # SEX=SEX # RACE=RACE # COUNTRY=COUNTRYNAME # ETHNIC=ETH # AGE=AGE)
USUBJID	opt., concatenating STUDYID-SITEID-SUBJID by default, or concatenating STUDYID-SUBJID if SUBJID already containing SITEID. CRF collected for extension study	%str(STUDY-SITEID-PTNO), %str(STUDYID-SUBJID)

autoSDTM Driven by SDTMIG

Example of %DM(continuing)

Parameter	Definition/Input Format	Example
RFSTDTC	req., DSNM.DATVAR|DSNM.TIMVAR, providing source var for RFSTDTC derivation/definition (min)	%str(EX.EXSTDAT|EX.EXSTTIM)
RFENDTC	req., DSNM.DATVAR|DSNM.TIMVAR, providing source var for RFENDTC derivation/definition (max)	%str(EX.EXENDAT)
RFXSTDTC	req., DSNM.DATVAR|DSNM.TIMVAR, providing source var for RFXSTDTC derivation (min)	%str(EX.EXSTDAT|EX.EXSTTIM)
RFXENDTC	req., DSNM.DATVAR|DSNM.TIMVAR, providing source var for RFXENDTC derivation (max)	%str(EX.EXENDAT)
RFPENDTC	req., DSNM.DATVAR|DSNM.TIMVAR, # separated for multiple, providing source vars (referring to flow chart in protocol) to derive RFPENDTC(max)	%str(EOS.EOSDAT # EOT.EOTDAT # AE.AESTDT # AE.AEENDT # CM.CMSTDT # CM.CMENDT # EX.EXENDAT)
DTHDTC	DSNM.DATVAR|DSNM.TIMVAR/WHERE, TIMVAR/WHERE optional. If the parameter not provided, LOG warning issued and DTHDTC populated as null	%str(SAE.DTHDAT)

autoSDTM Driven by SDTMIG

Example of %DM(continuing)

Parameter	Definition/Input Format	Example
BRTHDTC	YY=BRTHYY or YY=BRTHYY|MM=BRTHMO|DD=BRTHDD|TM=BRTHTIM (keyword: YY, MM, DD, TM) or DT=BRTHDAT|TM=BRTHTIM (keyword: DT, TM), TM added if collected	%str(DT=DM.DOB)
RFICDTC	req., format: DSNM.DATVAR	%str(IC.ICDT)
DMDTC	DSNM.DATVAR, if collected	%str(DMDAT)
SUPPDM	DSNM.QNAM/QLABEL='...', list of all non-standard variables collected from CRF, # separated if multiple, QLABEL optional	%str(DM.VISIT/QLABEL='Label of VISIT' # DM.VISITNUM)
NSUBMIT	DSNM.SRCVAR, list of [NOT SUBMITTED] variable if any	
ARM	DSNM.VARNM/WHERE, providing source var to assign ARM	%str(IC.REGIMEN/ELIGYN='Yes')
ACTARMCX	LIBNAME.DSNM, allocate Unplanned Treatment in ACTARMCD	%str(MISC.TRTCX)
STDCX	SDTMVAR/format=xxx, used for CT alignment if needed, # separated if multiple	SEX/format=$SEX
GENCODE	generating plain SAS codes if Y, N by default	%str(Y)

autoSDTM Driven by SDTMIG

Output from Example of %DM

OUTPUT	Note
DM	dm.xpt, dm.sas7bdat
SUPPDM	dm.xpt, dm.sas7bdat, if any SUPPQ mapped
DEFINE Template File	DM.xlsx/SUPPDM.xlsx, containing tabs of DATASETS, VARIABLES, CODELISTS 
Plain SAS Codes	dm.txt. For review purpose, the Plain SAS Codes dynamically generated from macro parameter settings 
LOG	DM.LOG, besides the SAS LOG displayed in SAS LOG window, it's also saved as a physical file 
Mapping and ANNOTATION SPECS	crfmappingspecs.sas7bdat for mapping review and autoACRF

VIEWTABLE: Sdtm.Crfmappingspecs

	RAWDS	RAWVAR	RAWVARLBL	DOMAIN	SDTMVAR	Specs	AnnoCRF
1	DM	AGE	Age	DM	AGE	DM.AGE	DM.AGE
2	ICPA	REGIMEN		DM	ARM	DM.ARM WHERE ELIGYN='Yes'	DM.ARM
3	DM	DOB	Date of Birth	DM	BRTHDTC	DM.BRTHDTC	DM.BRTHDTC
4	DM	COUNTRYNAME	CountryCode	DM	COUNTRY	DM.COUNTRY	DM.COUNTRY
5	DTH	DTHDAT	Date of Death	DM	DTHDTC	DM.DTHDTC WHERE PCDTH^=''	DM.DTHDTC
6	DM	ETH	Ethnicity	DM	ETHNIC	DM.ETHNIC	DM.ETHNIC
7	DM	RACE	Race	DM	RACE	DM.RACE	DM.RACE

autoSDTM Driven by SDTMIG

Keynotes to some other domains

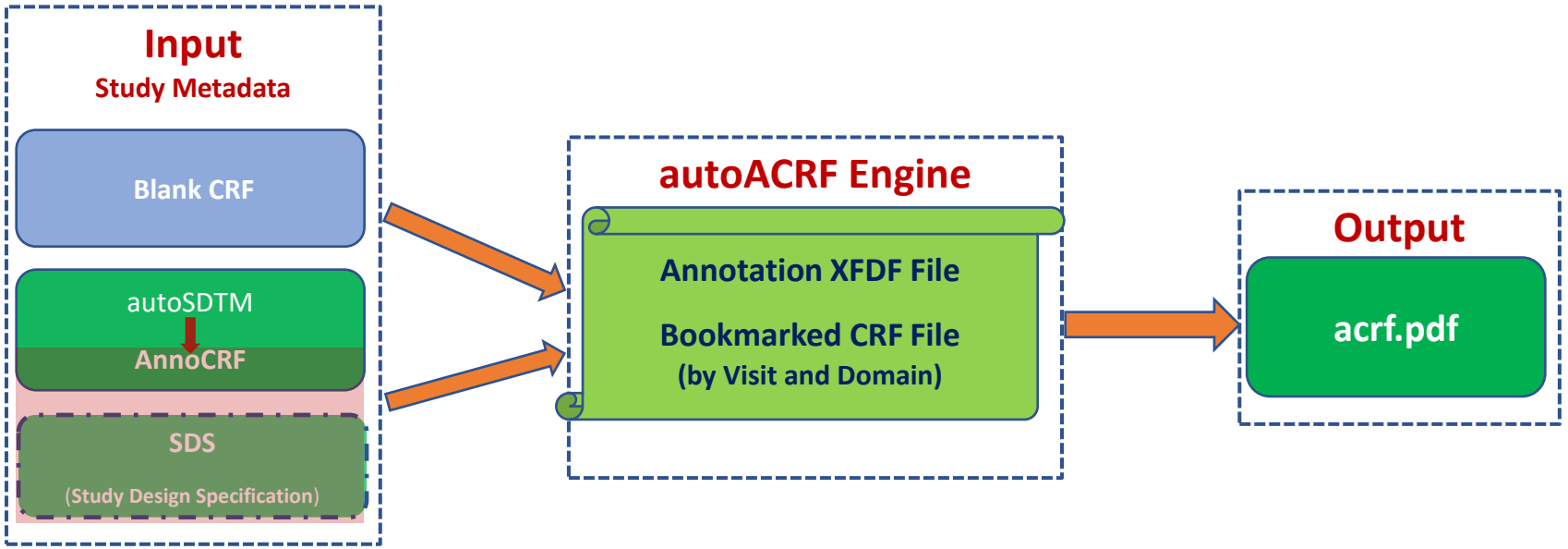
- SV as required, either CRF collected or derived
- SE as derived, aligned with TA, UNPLAN and FUP adjustment incorporated
- CM. CMDECOD further divided if >200 and CMDECODn automatically mapped to SUPPCM (WHO-DD B3-/C3- compliance)
- EC: SCHEDULED vs PERFORMED dynamically supported if ECMOOD CRF collected
- DS: by DSCAT ("DISPOSITION EVENT," "PROTOCOL MILESTONE," and "OTHER EVENT"), thus the mapping operation challenge resolved
- Finding domains (e.g.,VS), automatically supporting source data NORMALIZED vs DENORMALIZED (vertical vs. horizontal), currently FOUR patterns automatically supported
- RELREC mapping supported if CRF collection facilitated

….

autoACRF Streamlined Solution

Programming autoACRF Solution

- ❖Aligned with Study Metadata
- ❖Streamlined with autoSDTM



autoSDTM Driven by SDTMIG

Summary

- **Design goals met**
  - **Programming mapping solution**
  - **Universal solution** supporting any clinical data formats
  - **Transparent** to facilitate review and usage
    - ✓ Parameterizing and getting
    - ✓ Mapping specs and annotation specs dynamically produced
    - ✓ Plain SAS codes dynamically generated
  - **autoACRF Harmonized**
  - **Submission ready outputs**
  - **Easy operation** following programmer's way
    - ✓ User just listing source variable, then the macro handling all
    - ✓ UI/UX, feedbacks from programmers
- **Low transition cost**
- **High quality and efficiency**

Agenda

Introduction

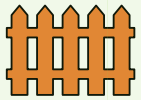
autoSDTM Driven by SDTMIG

autoADaM Driven by SAP

autoTLF Driven by ARM

Questions

autoADaM Driven by SAP

 **Safety autoADaM as good staring point**	❖Safety analyses more straightforward and based on actual observed data, less challenges compared with efficacy ❖Single SDTM domain as input dataset
 **Safety autoADaM implemented with GPS navigations**	❖ADAE (OCCDS), ADLB(BDS), ADEG(BDS), and ADVS(BDS)
 **Extended to other non-efficacy ADaM**	❖ADSL and ADCM also implemented with the same method applied ❖Similarly, ADMH also implementable
 **Limitation**	❖Currently limited to parallel study design supporting one or multiple treatment periods, easily extended to cross-over study
 **Efficacy autoADaM Development**	Under investigation and development ❖Two-layer design as a generic method ❖Endpoints further by TA or indication

autoADaM Driven by SAP

GPS Navigation Method

Global

- Global variables
  - ✓ Applicable to all projects and studies
  - ✓ Populated or derived by default without user input
  - ✓ Approximately 80% are global, fitting into 80/20 rule
  - ✓ Examples: RANDFL, SAFFL

Project

- Project variables
  - ✓ Applicable to all studies under the project
  - ✓ Dynamic depending on SAP
  - ✓ Controlled with macro parameters though user input
  - ✓ Examples: ASTDT/ASTDTF with imputations

Study

- Study variables
  - ✓ Applicable to the specific study
  - ✓ Dynamic depending on SAP
  - ✓ Controlled with macro parameters though user input
  - ✓ Examples: ADURN with end date imputations

Safety autoADaM with GPS navigation method, e.q., ADaM development driven by SAP, thus fully supporting SAP with analysis readiness, and global variables incorporated further achieving traceability

autoADaM Driven by SAP--User Experience

Statistical programmer as primary user and statistician as another key stakeholder

Easy to understand and easy to use, targeting to low transition cost

Platform			Multiple platforms are supported, e.g., PC SAS and EG SAS and extensible
	Parameter	Name	Straightforward and intuitive to understand
		Input	Explicit input or with a look-up table
		Rules or Conventions	Consistent input patterns across macros and/or parameters such as using “#” denoting multiple items. Macro to catch ERRORs (such as invalid input parameters) to avoid unpredictable SAS messages
	Performance		Optimize performance by using faster processing methods such as PROC SQL or PROC COPY. Use of slow procedures such as sorting will be limited to only when it is necessary.
 	Output	SAS LOG	Saved as an external file and replayed back to the SAS LOG Window
		Format	Use consistent formats across all datasets, e.g., all date in YYMMDD10. format
		Datasets	ADaM datasets stored in SAS and XPT formats
		Plain SAS Code	Plain SAS **executable** adxx.txt file dynamically generated based on the macro call instance. Besides used for code review, the plain SAS codes can be directly used for submission
		Specification	Specifications are dynamically produced based on the macro call instance. Specifications layout facilitates the use of define-xml tool
		Facilitate Review	Execution Summary Report generated in separate PDF and Excel files

autoADaM Driven by SAP

Example of %ADAE

Global variables

- SDTM AE variables
- Treatment date/datetime variables
- Treatment group variables including derived TRTA
- Ordering variables for ordinal variables, e.g., AESEV, AEREL
- Max Sev Variables, e.g., AOCCPIFL (1st Max Sev./Int. Occur Within PT Flag) and AOCCSIFL (1st Max Sev./Int. Occur Within SOC Flag)
- Global variables included for traceability being enabled

Design and implementation: %ADAE

P/S variables
(Macro parameters)

Parameter	Core	Definition
POPU	Optional	Analysis population set variable to be merged from ADSL onto ADAE Default: SAFFL
ADSL_VAR	Optional	Supportive/core variables from ADSL that will be merged onto ADAE Default: SITEID
STDT_IMPUT	Optional	Enable imputation of the AE onset date when the AE onset date is missing
ENDT_IMPUT	Optional	Select the imputation method for AE end date when the AE end date is missing. In order to derive ADURN, the AE end date imputation method must be defined. Default: AE end date imputation method is not defined; thus, ADURN is not derived
ATTR_IMPUT	Optional	AE attribute variable imputation list, e.g., AESEV # AEREL
GROUPING	Optional	Grouping variable definition using a SAS format, ordering variable derived by default
LAG	Optional	Define the number of days used as the lag for the treatment phase If a value of 0 is entered, then there will be no lag. If “MAX” is entered, then the time until the next treatment period will be considered as part of the current treatment period, and follow-up phase considered as last treatment period. Default: 0
SUPPQ	Optional	List the QNAMs to be merged into ADAE from SUPPAE

autoADaM Driven by SAP

Summary

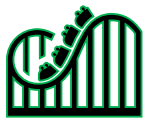
- Safety-based autoADaM successfully implemented with SAS macros
  - ADAE, ADLB, ADEG, ADVS
  - GPS navigation method applied, correspondingly, converted to SAP driven
  - Outputs with submission readiness
  - Specs and plain SAS executable codes dynamically generated from macro call
  - Easy operation, UI/UX incorporated
- Same method extended to other non-efficacy ADaM, e.g., ADCM, ADMH
- High quality and efficiency with low transition cost
- Efficacy-based autoADaM under investigation and development

Agenda

Introduction

autoSDTM Driven by SDTMIG

autoADaM Driven by SAP



autoTLF Driven by ARM

Questions

autoTLF Driven by ARM

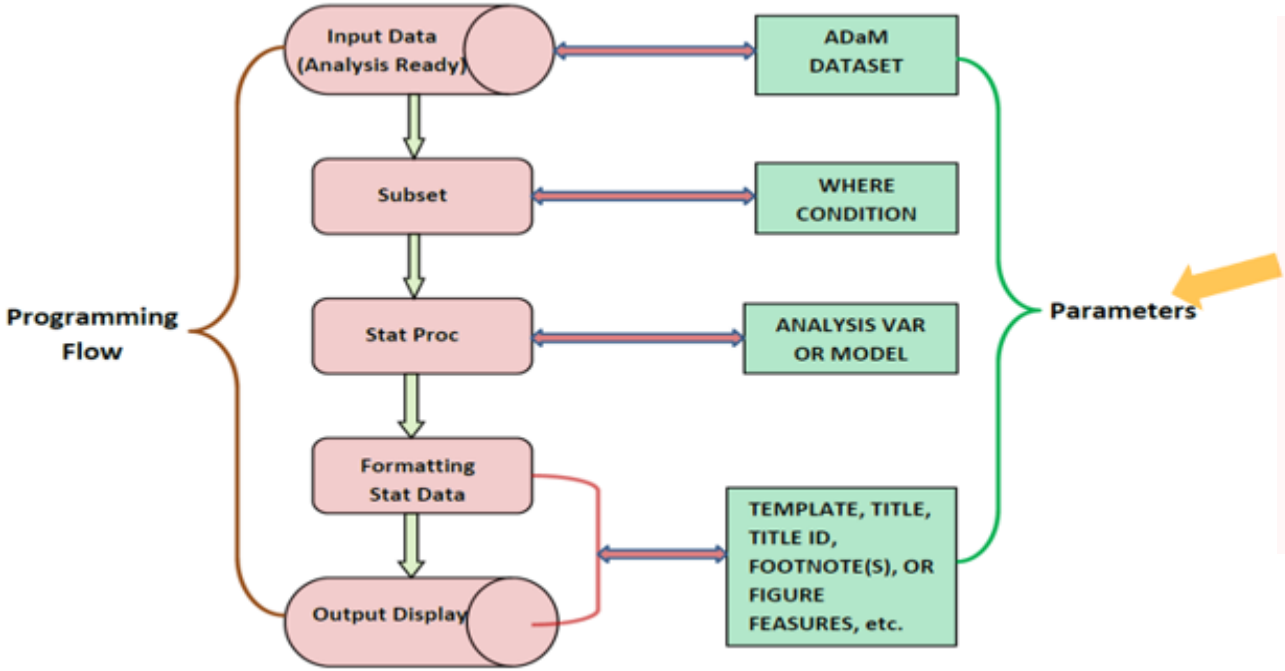
Analysis Results Metadata (ARM) introduction

- Three types of metadata defined in ADaM v2.1 (Dec. 2009)
  - Analysis Dataset Metadata
  - Analysis Variable Metadata
  - **Analysis Results Metadata (ARM)**
- Analysis Results Metadata functioning as table specifications
  - Describes the main attributes of the analysis results
  - Provides link between analysis results and analysis dataset to facilitate reviews
- ARM v1.0 for Define-XML V2 supporting ARM submission published in Jan 2015
- ARM submission for primary/secondary endpoints and main safety analyses recommended and becoming common practice

autoTLF Driven by ARM

ARM applications

- ❖ Switching process from POST to PRIOR, functioning as table specifications, e.g., annotations on table template
- ❖ SAS macro parameters aligned with ARM fields, thus TLF generations driven by ARM
- ❖ With macro call instance transparent interface, review greatly facilitated for the analysis
- ❖ To achieve autoTLF, it's prerequisite with ADaM analysis-ready or pre-processing ADaM to be analysis-ready. After applying data selection criteria WHERE statement, it's able to directly invoke SAS statistical procedure



- association exists between programming process and ARM
- macro parameters design aligned with ARM fields
- thus, macro call instance as table specs as well
- additionally, ARM define-xml generation automated and harmonized with production

autoTLF Driven by ARM

Freq analysis with OCCDS, similarly to DSTAT with BDS

%macro FREQ(

1 SAP

TableID	=	Table number
Title	=	Table titles
Popu	=	Analysis population set
Footnote	=	User defined footnotes

2 Programming

Indata	=	Input analysis dataset
Where	=	Data selection criteria
SubGroup	=	Subgroup analysis variable
Analvar	=	Analysis variables, always counting WORST for *SEV
ADSL_TRT	=	Treatment period pair variables from ADSL
Total	=	Total or subtotal for the treatment group
FreqThreshold=		Frequency threshold minimum cut-off value for display

);

Industry Production Level Application

1. Very powerful for any frequency tables
2. ARM driven as framework extended to frequency analyses (n, %)
3. 4 sort options per var enabled, including complete types (if data not a.v. for one category, still displayed with 0 filled in)
4. Some inferential statistic, e.g., RD, may be further added via additional parameter

autoTLF driven by ARM

ARM management in production

- ❖Macro parameters
- ❖Part of ToC
- ❖Specification fields with reporting system

ARM define-xml automated and aligned with production

Solution implemented and published in conferences (PharmaSUG 2015, CDISC Interchange 2015, DIA 2017, and PharmaSUG 2021)

Upgrading inferential analysis macros (MMRM, ANCOVA, LIFETEST, CMH, …) with ARM driven method to industry production level

Summary

End-to-end applicable autoProg solutions available

- ❖autoSDTM (including autoACRF), autoADaM, and autoTLF
- ❖High efficiency, high quality, and low transition cost
- ❖Outputs with submission readiness
- ❖Plain SAS codes autogenerated based on the macro call instance
- ❖Keep the solution simple

Mission to reshape current resource intensive programming world

- ❖Resolving production pain points
- ❖Products (macros, reporting system)
- ❖Technical supports
- ❖Further development by integrating UI and UX

Questions

Chengxin(Cheng) Li
AutoCheng Clinical Data Services LLC
Warren, NJ
Chengxin.li@autoclindata.com
Cell: 862-308-3089