phuse·hybrid

Single Day Event

Ram Mohan Konda,
Navitas Data Sciences

Automation by Macro's in SAS programming

phuse.global

Why Automation?

- Faster way to generate results
- Meeting High-quality every time
- Applicable to datasets, variables, reports
- Validation efforts can be reduced
- End user need not have core programming knowledge
- Cost effective

Benefits of Automation

Reduce Workload:

We have some unique tables and listings with respective ADAE, ADMH, and ADCM datasets, etc. By automating unique tables and listings using macros, the repetitive task can be avoided. Often In SDTM or ADaM datasets, programmers need to derive %EPOCH, %SEQ, and %DUR. Deriving these in each domain is a time-consuming, uniform across all domains and repetitive process.

Reduce Potential Errors:

Individual Programmer may miss some check points which may impact the first-time quality. By automation we can reduce the risk ratio.

Automation by Macro's in SAS programming

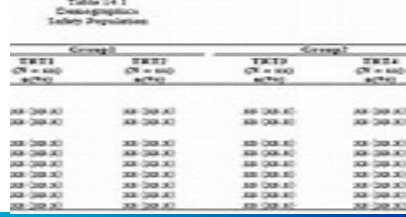
Control:

By controlling the input dataset variable, we can produce the required output

Time Management:

By automation we can complete more work within short period of time.

Concomitant Medication Macro (%cmtab) Case Study:



Objective: To Create a Summary Table of Concomitant Medications by Therapeutic Class and Preferred Name by Population by using %cmtab macro

Draft Mock-Up

Summary of Concomitant Medications by Therapeutic Class and Preferred Name – (Population: DBT Safety Sample)

Therapeutic Class **Preferred Term**	**Treatment A (N=xx) n (%)**	**Treatment B (N=xx) n (%)**
Subjects with Concomitant Medication	xx (xx.x)	xx (xx.x)
CMCLAS 1	xx (xx.x)	xx (xx.x)
PT 1	xx (xx.x)	xx (xx.x)
PT 2	xx (xx.x)	xx (xx.x)
CMCLAS 2	xx (xx.x)	xx (xx.x)
PT 1	xx (xx.x)	xx (xx.x)
PT 2	xx (xx.x)	xx (xx.x)
...		

Concomitant Medication Macro (%cmtab)

Step 1:

To check the input data set is not available.

- when the input data set is not available then the macro will write the log message.

" Warning: Analysis.ADCM dataset is not available."

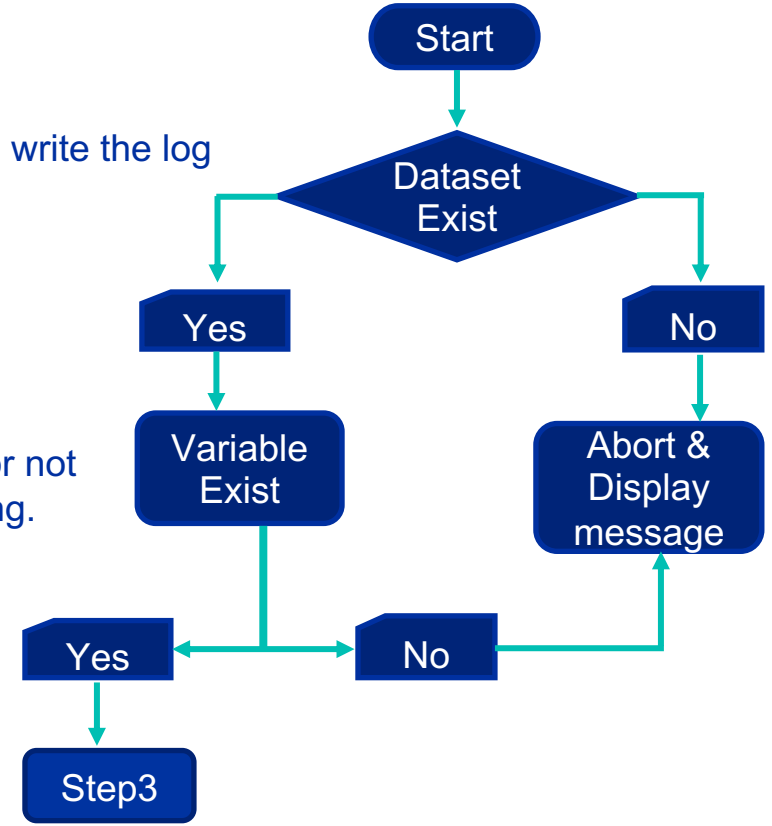
%abort;

Step 2:

- To check the required variable are present in the dataset or not
- if any variable missing in the dataset, it shows a log warning.

"Warning: &varchk variable is missing in the dataset."

%abort;

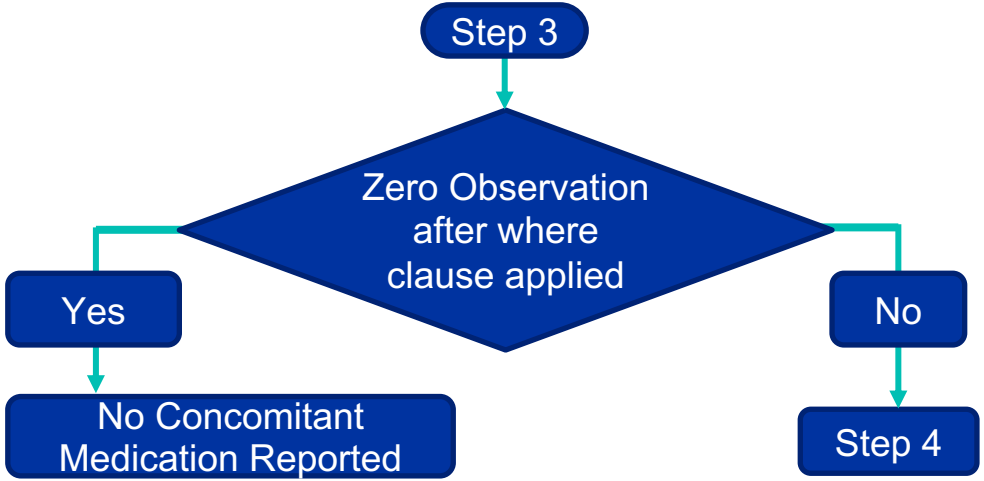


Concomitant Medication Macro (%cmtab)

Step 3:

- when Zero observation after where clause applied is then macro will write the log message.

The report needs to be displayed as
"No Concomitant Medication Reported"
%abort;



Step 4:

- Macro variable to the maximum number of Treatment Counts
- Macro variables to store Numeric and Character Treatment values
- Macro Variables to store the Header count of Each treatment

Usage of Numeric and Character treatment values and corresponding counts in Proc Report

Column: To create a Column header for each treatment group.

Define: To create Column Labels with population counts.

Concomitant Medication Macro (%cmtab)

Step 5: To create a total treatment in ADSL and ADCM datasets and the total column is controlled by macro parameters

Step 6: Creation of column header count dataset for calculating the percentage

Step 7: Avoid duplicate records with respective frequency counts.

Creation of Frequency counts for

a) Any Medication with respective treatments b) TRT01A*CMCLAS c) TRT01A*CMCLAS*CMDECOD

Step 8: Appending all frequency datasets & Transpose the percentages with numeric treatment values

Step 9: To align the dataset based on the mock shell.

Step 10: Reporting

Concomitant Medication Macro (%cmtab)

Exceptional Handling

1) Analysis dataset is not available.
2) No dataset name for reporting is supplied
3) Variable is missing in the dataset
4) Nowhere clause supplied for use with the dataset
5) Zero Observations in ADSL after WHERE CLAUSE applied
6) Treatment Group with no Subjects

Total Number of Studies: Multiple

Manual Time for the Programmer

a) Program Development: 40 hours b) Unit Test: 20 hours
c) Efforts saved in repetitive studies: 3 hours per unique table

Advantage: 1) Even, We can use %cmtab macro for Medical history and Adverse Events tables.
2) We can generate reports irrespective of the number of treatment

EPOCH Macro (%EPOCH)

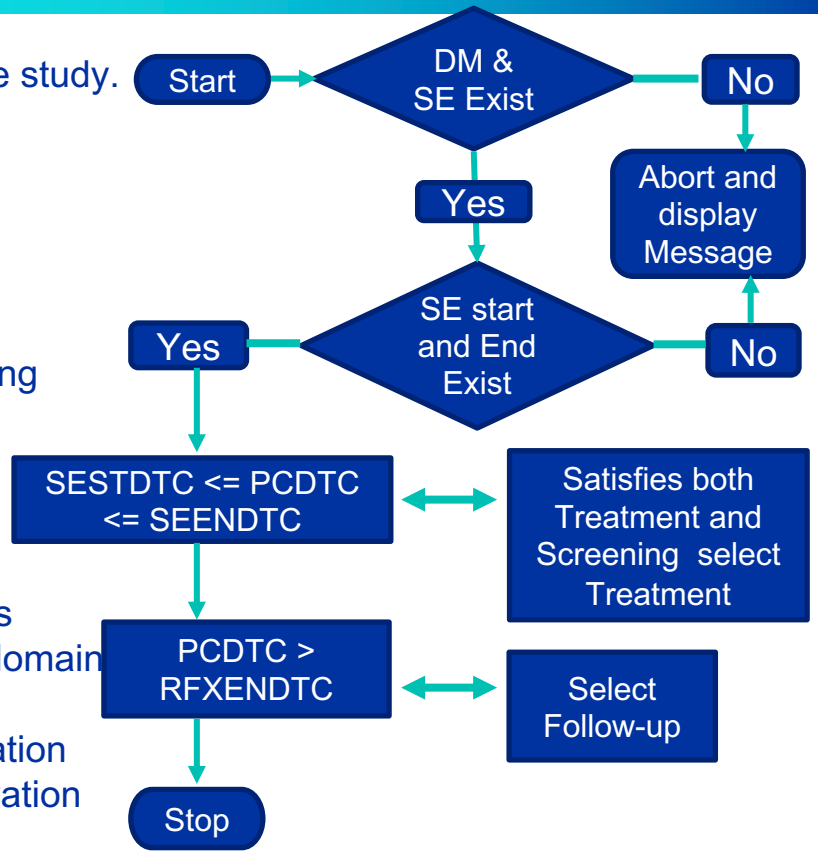
STUDYID	USUBJID	PCSEQ	EPOCH	PCDTC
XXXX	0001	1	Screening	2015-12-04T14:02
XXXX	0001	2	Follow-up	2015-12-08T13:56
XXXX	0002	1	Screening	2015-12-14T10:36
XXXX	0002	2	Treatment	2016-03-07T09:28
XXXX	0002	3	Treatment	2016-03-07T10:40
XXXX	0002	4	Treatment	2016-03-07T11:54

Objective: Epoch timing periods are by definition during the study.
Number of Studies: Multiple
Domains Used: AE,PC,CM,DS,VS,LB,EX,PE,TA,EG,BE

Exceptional Handling:
1) DM and SE datasets do not exist
2) SESTDTC and SEENDTC variables do not exit or missing
3) XXDTC contains all missing values
4) RFXENDTC does not exist

Manual Time for the Programmer
a) Program Development: 40 hours b) Unit Test: 20 hours
c) Efforts saved in repetitive studies: 3 hours per unique domain

Advantages: To have a uniform approach for Epoch derivation In multiple studies. Interpersonal variability for epoch derivation can be eliminated.



Sequence number Macro (%SEQ)

STUDYID	USUBJID	DSSEQ	DSTERM	DSDECOD	DSSTDTC
XXXX	0003	1	PGt	INFORMED	
XXXX	0003	2	CSF	INFORMED	2016-03-28
XXXX	0003	3	Informed	INFORMED	2016-03-28
XXXX	0003	4	Per	SCREEN	2016-03-31

Objective: Calculates sequence numbers.
No of Studies: Multiple
Domains Used: AE,PC,CM,DS,VS,LB,EX,PE,TA,EG, PP

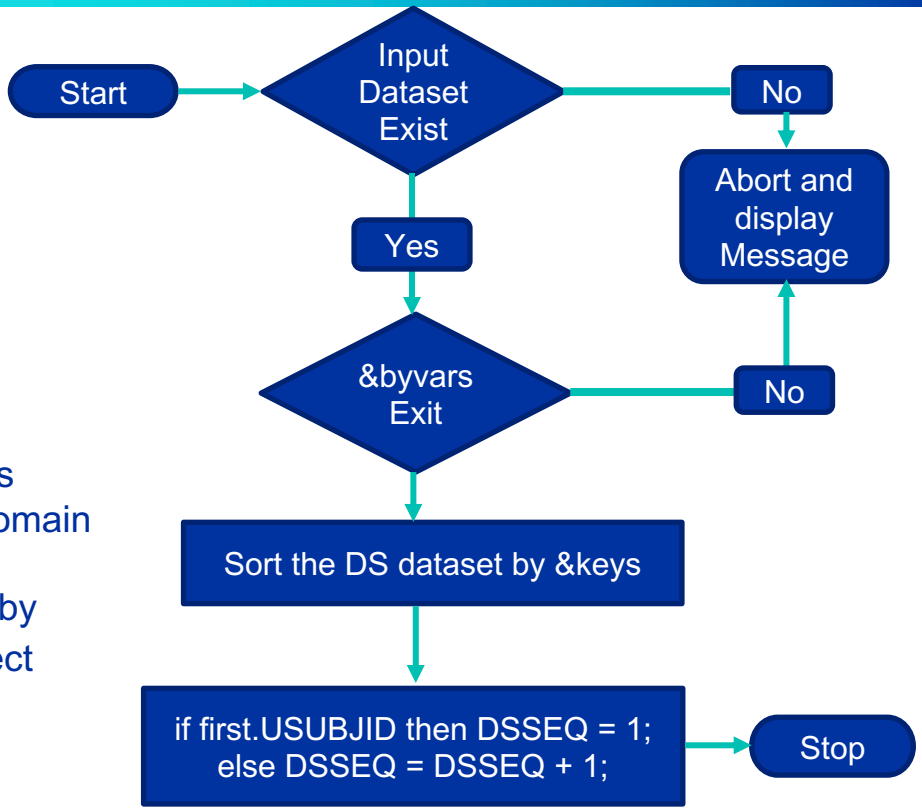
Exceptional Handling:

1) Input datasets are not available
2) &byvars variables do not exist

Manual Time for the Programmer

a) Program Development: 20 hours b) Unit Test: 10 hours
c) Efforts saved in repetitive studies: 10 mins per unique domain

Advantage: Once SDTM or ADaM specification is updated by Statisticians related to key variables automatically will reflect In the dataset after re-run



Duration Days Macro (%DUR)

STUDYID	USUBJID	DSSTDTC	DSDY
XXXX	0001	2015-08-17	-110
XXXX	0001	2015-09-14	-82
XXXX	0001	2015-10-19	-47
XXXX	0001	2015-12-04	-1
XXXX	0001	2015-12-04	-1
XXXX	0001	2015-12-08	4

Objective: **C**alculates study day.
No of Studies: Multiple
Domains Used: PC,DS,VS,LB,PE,MH,EG,BE, ADSL, ADMH

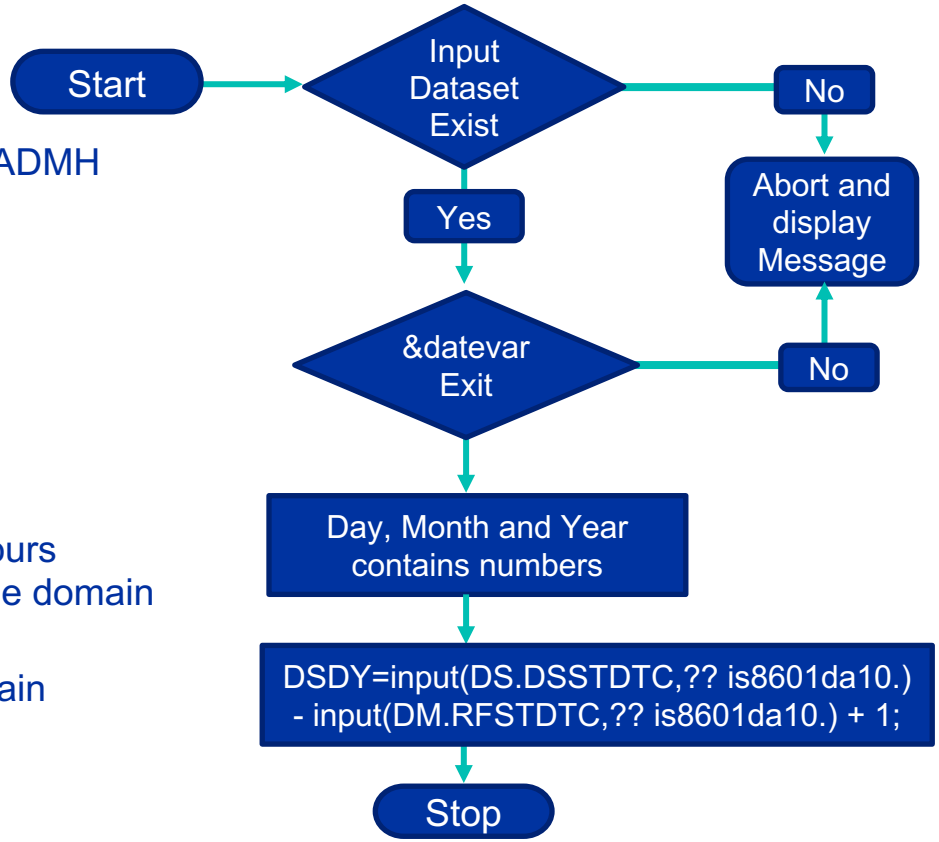
Exceptional Handling:

1) Input datasets do not exist
2) Date variables are missing or do not exist
3) RFXENDTC variable does not exist

Manual Time for the Programmer

a) Program Development: 20 hours b) Unit Test: 10 hours
c) Efforts saved in repetitive studies: 10 mins per unique domain

Advantage: Macro will not calculate if date values contain Character values and it will flag a warning.



The Efforts involved in the Validation of Macros

1) Validated whether correct results are generated with the known data and known results.

2) Validated credible results generated with the unknown data.

3) Validated all the variations in input parameters and verify their effects.

4) Validated the erroneous data and check whether the expected error and warning messages are generated.

5) Checked for grammar and spelling errors in comments.

6) Checked if all the temporary SAS variables created by the macro in the temporary dataset have a unique name.

7) Tested readability, readable output layout (tables, reports), and standard comments on each output page with date, SAS version, macro name, etc.

8) Checked whether the comments are generated in the log file for optional debug system options.

9) Write a validation report, both for approval or disapproval.

Summary

1. We chose a simple example of creating a table for ease of demonstration.

2. %cmtab macro used for CM summary mock shell (as illustrated) also resembles AE & MH Summary table so this macro's scope can be extended for unique AE and MH Summary table as well.

3. Using macros to check whether the dataset is available or not for the table.

4. Using macros and checking whether the variables are present in the dataset or missing.

5. Storing all the treatment values into a table using single-step macros.

6. Submit the table as per the specified mock shell using Proc Report.

7. Finally, Efforts are involved in the validation of the Macro.

8. Deriving EPOCH, _ _SEQ, and _DURATION calculation using macros.

Conclusion

1. SAS Automation is a reliable and efficient method for repetitive programming tasks. Higher efficiency will also lead to better quality.

2. We have learned how to create tables by controlling the input dataset

3. All the scenarios that might occur need to be brained stormed before we start with automation

4. Macro should be universal to fit different study designs and layouts

5. Maintain proper documentation/usage guide to make it simpler for the end user

6. Utility Macros used in the presentation will significantly cut down the programming time.

phuse

The Global Healthcare Data Science Community

Contact Channels

UK +44 1843 609600
office@phuse.global
phuse.global

Social Media

@phusetwitta
/phusebook
/phusetube
/company/phuse

Thank You