PharmaSUG 2013 — Paper DS-02

Leveraging SDTM Standards to Cut Datasets at Any Visit

Anthony L. Feliu, Genzyme, A Sanofi Company, Cambridge, Massachusetts
Stephen W. Lyons, Genzyme, A Sanofi Company, Cambridge, Massachusetts

ABSTRACT

Clinical trials with complex design or long duration often include interim milestones for evaluation of safety and efficacy. Since clinical databases are seldom configured to deliver other than full extracts, the responsibility for subsetting data invariably falls to the statistical programmer.

In this paper, the authors walk through the process of specifying, programming, and validating a generic macro to cut SDTM datasets at any scheduled visit. Three considerations dominated our ultimate design: (a) the tool could be used on any protocol; (b) the tool would first determine the timing variables present a given domain and then apply appropriate cascade logic; and (c) a diagnostic footprint would allow for human review of the decisions made by the tool.

INTRODUCTION

Given widespread adoption of CDISC standards for data tabulations (SDTM), programmers have the opportunity to step back from specific protocols and develop generic solutions to common programming tasks.

SDTM datasets are an ideal launch pad for solutions building because clinical data are neatly and predictably formatted into any of several “domain classes”. Most data fall into the interventions, events, and findings domain classes, while the special–purpose and trial design classes hold supporting data.

The variables comprising any given domain are prescribed by the standard, which includes an explanation of their purpose, or “role,” in communicating the data. Broadly speaking, a domain will consist of one topic variable, several identifiers (study id, subject id, etc), timing variables (visit designation, dates, time points), and various qualifiers. Qualifiers are further detailed as “result qualifiers” (holding the answer from a test), “variable qualifiers” (companion variables giving more information about their sibling; a units variable supports the result value, for example), and “record qualifiers” (information elaborating the topic; for example, a method variable explaining the test technique), among others.

Given the assignment to cut data for several phase 3 protocols, we developed a single macro that could operate on all SDTM datasets.

To contemplate cutting the data, timing is everything. These variables include:

- Visit designation (VISITNUM and VISIT) – Planned clinical encounter within the study schedule.
- Collection date (—DTC) – Occurrence of an instantaneous observation.
- Starting date (—STDTC) – First occurrence of the event or treatment, beginning of an observation.
- Ending date (—ENDTC) – Last occurrence of the event or treatment, completion of an observation.
- Study day (—DY, —STDY, —ENDY) – Elapsed days compared to reference date.

Unfortunately, the selection of timing variables not only differs between domain classes, but also among domains within a given class.

From this brief treatise on the SDTM standard, two requirements emerge for the envisioned data cutting macro:

- A generic macro will require the capability to identify the timing variables present in a given domain and apply the corresponding algorithms;
- Based on the specified cutoff visit, the macro is expected to deduce a subject–specific cutoff date to accept or reject records in domains which do not have visit information.

THE APPROACH

Our data cutting technology was developed through a formal requirements process.

First, a set of **use cases** was developed to describe how subjects can participate in a clinical trial.

Then, **business requirements** were drafted with minimum jargon to distinguish among these experiences, and test records for acceptance into the data cut.

From this analysis, **technical specifications** were prepared to fill in details about how the code would be written. At the same time, metadata the macro would need to execute its record selection algorithms were identified.

Next, prototype code was written and run with test datasets. The outputted data were reviewed iteratively by team programmers and statisticians to refine the specs.

Once satisfactory results were obtained from the prototype code, the requirements were given to a second programmer for blind parallel programming. Updates to the requirements and macro code were implemented to have a stable and well–documented process.

USE CASES

We sought to understand the range of subject experiences that may occur during a clinical trial and whether those data should be included in our cut. See Table 1.

Status	Case	**Situation**	**What to do**
Subject On Study	A	Subject did not yet reach the cutoff visit.	Accept all data
	B	The cutoff visit is the subject’s last clinical encounter. • Date of cutoff visit CRF is the latest visit date for the subject.	Accept all data
	C	Subject had planned or unplanned clinical encounter(s) beyond the cutoff visit. • Date of cutoff visit CRF is not the latest visit date for the subject.	Cut the data
Subject Off Study	D	Subject withdrew prior to cutoff visit • Did not reach cutoff visit, but completed end–of–study (EOS) visit.	Accept all data
	E	Subject withdrew at cutoff visit. The site should complete EOS CRFs instead of the regular visit CRFs, but suppose it submits both. • If date on EOS CRF equals date on cutoff visit CRF, withdrawal occurred at the cutoff visit.	Accept all data
	F	Subject withdrew after cutoff visit but prior to next planned visit. • If the date on EOS CRF is after the date on cutoff visit CRF, withdrawal occurred after the cutoff visit.	Cut the data
	G	Subject had planned or unplanned clinical encounter(s) beyond the cutoff visit and then withdrew. • Same as Use Case C. • The withdrawal event does not impact record selection.	Cut the data
	H	Subject completed the study per protocol. • Same as Use Case C.	Cut the data

Table 1. Use Cases for Record Selection

We quickly realized that data cutting depended on more complete information about a subject’s participation in a trial than what would be found in any one dataset. For this reason, we would be obligated to prepare a reference table with visit information, and use this for cutting the datasets.

We further concluded that our algorithms required particular care to evaluate whether unscheduled visits fell inside or outside the data cut. To do that, our programming would need to operate with both visit criteria and date criteria.

We were also challenged to recognize precisely when an End–of–Study (EOS) visit occurred. After all, an End–of–Study visit can occur at any point in the study timeline.

BUSINESS REQUIREMENTS

For an ongoing study, it is inevitable that some collected data will be incomplete or unclean. When there was uncertainty about the timing of a particular data record, our preference was to include it in the data cut.

First Tier Cascading Logic: Dataset–level

- RULE 1. When a domain has no timing variables, accept all records for all subjects.

Second Tier Cascading Logic: Subject–level

Compare a subject's progress in the study, evidenced by the reference table of visits, with the cutoff parameters. Determine whether all data can be accepted for the subject, or if a cutoff date is applicable.

- RULE 2. If a subject did not reach the cutoff visit, accept all records for that subject (Use Case A). If present, include End–of–Study records regardless of date (Use Case D).
- RULE 3. If the cutoff visit is the last visit for a subject, accept all records for that subject (Use Case B).
- RULE 3B. If a subject discontinued and the date for End–of–Study CRFs matches date for the cutoff visit, accept all records for that subject (Use Case E).

Third Tier Cascading Logic: Record–level

When a subject progressed beyond the cutoff visit, derive a cutoff date as follows:[1]

- If the subject attended the planned visit following the cutoff visit ("next visit"), set the cutoff date to be one day before the next visit. (Use Cases C, G, H)
- If the subject skipped the "next visit" but had later visits, impute the cutoff date to be cutoff–visit–date plus 7 days. (Use Cases C, G, H)
- If the subject had their End–of–Study (EOS) visit following their cutoff visit, set the cutoff date to be one day before the EOS visit date. (Use Case F)

Once the cutoff date has been derived, proceed with record–level logic.

RULE 4. When a domain has VISITNUM, accept all records for planned visits up to and including the cutoff visit.

RULE 5. When a domain has a start date variable (—STDTC[2]), compare the subject's cutoff date with the --STDTC date. Accept all records for that subject through the cutoff date.

RULE 6. When a domain has a date of collection variable (—DTC), apply similar tests to the —STDTC variable. But if the domain has both start date (—STDTC) and date of collection (—DTC) variables, apply Rule 6 only when STDTC is missing and --DTC is known.

RULE 7. If the domain has one date variable (—DTC or —STDTC) and it is missing, accept the record. If a domain has two date variables (—DTC and —STDTC) and both are missing, accept the record. <u>However</u>, do not accept records with missing dates when the domain has the VISITNUM variable.

Apply record selection using visit criteria (Rule 4) and date criteria (Rules 5–7) independently. That is, accept a record under the visit criteria even if the record date (—STDTC/—DTC) would exclude it, and vice versa. Independent record selection minimizes the possibility of rejecting records with incomplete or incorrect timing information.

---

[1] Flexibility in setting the cutoff date allows for labs or radiology performed after presenting to the investigator.

[2] Em-dash in a variable stem represents the domain abbreviation. The macro had to be specific to avoid capturing supplemental qualifier variables, which are non-standard and thus cannot be analyzed.

TECHNICAL REQUIREMENTS

The macro is expected to have these capabilities in order to operate on the datasets:

- The macro is responsible to detect the variables present in the given dataset, and apply the appropriate logic. The input dataset must have DOMAIN and USUBJID variables. Other variables of interest are optional: visit number (VISITNUM), visit name (VISIT), and date (—STDTC and/or —DTC).

- Character dates —DTC and —STDTC are converted to numeric dates. A missing day is imputed to first of the month, a missing month to January. But no imputation is performed for missing values.

- The macro's action can be controlled by parameter ("F" to flag records, "D" to delete records).

- When the macro action is set to flag records, two diagnostic variables are added to the output dataset:

  _FLG will indicate record acceptance (_FLG = 1) or rejection (_FLG = Null).

  _RULE with the rule numbers under which a record is accepted (1–7).

The macro will receive descriptive metadata about the study through a set of parameters:

• List of planned visit numbers falling within the data cut.	VISITNUM_PLAN
• The latest acceptable visit in this list being the cutoff visit.	VISITNUM_CUT
• Visit number of the first–occurring post-cutoff visit, the so called "next visit".	VISITNUM_NEXT
• Visit number for the study termination visit, "EOS visit".	VISITNUM_EOS
• A reference "visits" dataset with encounter dates for all subject visits. Four columns—USUBJID, VISITNUM, VISIT, DVDT (visit date). Our EDC downloads included a visits dataset, from which we readily generated this reference table with every data refresh. Were this not available to us, an alternative would have been to troll one or several raw data panels (vital signs, for example) to derive the visit information.	VISIT_DATASET

GENERIC MACRO SOLUTION

The standard programming process at our company is to parallel program all datasets. For this reason, two macros were independently written. %CUTDATA is for use by production programs, and %QCCUTDATA for validation. Each version checks its running environment and aborts if mistakenly called.

MACRO INVOCATION

Macro variables with metadata for the data cut were added to the project "setup" file. The action parameter (F flag records, D drop records, blank to do nothing) is thereby centrally controlled.

```
%let VISITNUM_PLAN  = 1 2 3 6 9 12 ;   /* Planned visits within the data cut.    */
%let VISITNUM_CUT   = 12 ;             /* Cutoff visit number.                   */
%let VISITNUM_NEXT  = 18 ;             /* First planned visit after cutoff.      */
%let VISITNUM_EOS   = 99 ;             /* Termination visit number.              */
%let VISIT_DATASET  = ref.dvx          /* Visit information for all subjects.    */
%let CUTDATA_ACTION = F ;              /* F = flag records. D = delete records. */
```

Team programmers were then instructed to add a call to the %CUTDATA macro in their SDTM programs:

```
%CUTDATA (dsin      = work.lb ,          /* Input SDTM dataset.  */
   dsout            = work.lbcut ,       /* Output SDTM dataset. */
   visitnum_plan    = &visitnum_plan ,
   visitnum_cutoff  = &visitnum_cut ,
   visitnum_next    = &visitnum_next ,
   visitnum_eos     = &visitnum_eos ,
   visit_dataset    = &visit_dataset ,
   action           = &cutdata_action ) ;
```

Validation macro %QCCUTDATA has an identical set of parameters.

DETERMINATION OF SUBJECT–LEVEL "CUTOFF DATE"

Cutting the data requires more complete visit information than might be found in the individual SDTMs. For this reason, a reference dataset is generated from each download of clinical data. Using this reference dataset, the %CUTDATA macro applies second–tier logic (see above) to determine if all records should be accepted for the subject, or if a cutoff date is applicable.

Table 2 presents a sample print with four subjects.

Obs	USUBJID	VISIT–NUM	DVDT	VISIT
1	0000-0001	1	2008-09-29	Screening
2	0000-0001	2	2008-10-13	Day 1
3	0000-0001	3	2009-01-12	Month 3
4	0000-0001	6	2009-04-13	Month 6
5	0000-0001	88	2009-04-20	Unscheduled
6	0000-0001	9	2009-07-13	Month 9
7	0000-0001	12	2009-10-05	Month 12
8	0000-0001	18	2010-04-12	Month 18
9	0000-0001	24	2010-10-04	Month 24
10	0000-0001	88	2010-10-19	Unscheduled
11	0000-0001	30	2011-04-10	Month 30
12	0000-0001	99	2011-05-29	End of Study
13	0000-0002	1	2008-10-19	Screening
14	0000-0002	2	2008-11-01	Day 1
15	0000-0002	3	2009-01-29	Month 3
16	0000-0002	6	2009-01-01	Month 6
17	0000-0002	99	2009-06-23	End of Study
18	0000-0003	1	2008-06-07	Screening
19	0000-0003	2	2008-06-20	Day 1
20	0000-0003	3	2008-09-19	Month 3
21	0000-0003	6	2008-12-17	Month 6
22	0000-0003	9	2009-03-22	Month 9
23	0000-0003	12	2009-06-22	Month 12
24	0000-0003	99	2009-07-06	End of Study
25	0000-0004	1	2008-07-17	Screening
26	0000-0004	2	2008-07-30	Day 1
27	0000-0004	3	2008-10-29	Month 3
28	0000-0004	6	2009-01-26	Month 6
29	0000-0004	9	2009-05-01	Month 9
30	0000-0004	12	2009-08-01	Month 12
31	0000-0004	99	2009-08-01	End of Study

Table 2. Sample print of reference dataset with subject—visit information

Assuming the earlier–mentioned invocation parameters are in effect, the %CUTDATA macro will make these conclusions:

- **Subject 0001** — All planned visits were completed.

Cutoff visit	Month 12	2009-10-05
"Next" planned visit	Month 18	2010-04-12
End of Study	EOS	2011-05-29 (after "Next" visit date)

Cut the data. Cutoff date will be "next" visit minus 1 day = 2010-04-11

- **Subject 0002** — Prematurely discontinued.

Cutoff visit	Month 12	Not done
"Next" planned visit	Month 18	Not done
End of Study	EOS	2009-06-23

Accept all records. Cutoff date is not applicable because subject did not progress in trial.

- **Subject 0003** — Prematurely discontinued.

Cutoff visit	Month 12	2009-06-22
"Next" planned visit	Month 18	Not done
End of Study	EOS	2009-07-06

Subject had cutoff visit but discontinued before the "next" visit.
EOS date is after cutoff visit date.

Cut the data. Cutoff date will be EOS date minus 1 day = 2009-07-05.

- **Subject 0004** — Prematurely discontinued.

Cutoff visit	Month 12	2009-08-01
“Next” planned visit	Month 18	Not done
End of Study	EOS	2009-08-01 (EOS date = cutoff visit date)

  Accept all records. Cutoff date is not applicable because subject discontinued at the cutoff visit.

DATA CUTTING RESULTS

DOMAIN WITH VISIT AND DATE (TABLE 3)

A sample lab recordset for subject 0001 is presented in Table 3.

- Records 1–6 were accepted under Rule 4 because their visit designations were within the data cut (&VISITNUM_PLAN).
- Records 7–12 do not qualify for acceptance by visit designation (Rule 4), but they had a second chance to be evaluated by date.

  “Month 18” was rejected even though LBDTC is missing. (Rule 7 allows records with missing dates to be accepted only when a domain lacks visit information.)

  Unscheduled record 10 was accepted because LBDTC is earlier than the cutoff date 2010-04-11 (Rule 6).

  Unscheduled record 11 and EOS record 12 were rejected by Rule 6 because LBDTC after cutoff date.

Obs	DO–MAIN	USUBJID	LBTESTCD	VISIT–NUM	VISIT	LBDTC	LB–DY	LBSTAT	_FLG	_RULE
1	LB	0000-0001	SODIUM	1	Screening	2008-09-29T13:15	-14		1	4
2	LB	0000-0001	SODIUM	2	Day 1			NOT DONE	1	4
3	LB	0000-0001	SODIUM	3	Month 3	2009-01-12T10:12	92		1	4
4	LB	0000-0001	SODIUM	6	Month 6	2009-04-13T11:34	183		1	4
5	LB	0000-0001	SODIUM	9	Month 9	2009-07-13T12:20	274		1	4
6	LB	0000-0001	SODIUM	12	Month 12	2009-10-05T10:53	358		1	4
7	LB	0000-0001	SODIUM	18	Month 18			NOT DONE		
8	LB	0000-0001	SODIUM	24	Month 24	2010-10-04T11:38	722			
9	LB	0000-0001	SODIUM	30	Month 30	2011-04-10T08:55	910			
10	LB	0000-0001	SODIUM	88	Unscheduled	2009-04-20T10:53	190		1	6
11	LB	0000-0001	SODIUM	88	Unscheduled	2010-10-19T08:50	737			
12	LB	0000-0001	SODIUM	99	End of Study	2011-05-29T13:40	959			

Table 3 Sample SDTM with both visit and date variables

DOMAIN WITH DATE ONLY (TABLE 4)

For domains without visit information, data cutting is entirely dependent on the subject–level determination either to accept all data or to apply a cutoff date.

Table 4 presents sample adverse event records for Subject 0003. Recall that for this subject, cutoff date 2009-07-05 is applicable.

Obs	DOMAIN	USUBJID	AESEQ	AETERM	AESTDTC	AEENDTC	AESTDY	AEENDY	_FLG	_RULE
1	AE	0000-0003	1	ABDOMINAL PAIN		2008-06			1	7
2	AE	0000-0003	2	DIARRHEA	2008-06-21	2008-06-22	2	3	1	5
3	AE	0000-0003	3	TICK BITE	2008-06-29	2008-06-29	10	10	1	5
4	AE	0000-0003	4	RIGHT KNEE INFLAM	2008-12-20		184		1	5
5	AE	0000-0003	5	RIGHT ELBOW PAIN	2009-03-06	2009-04-06	260	291	1	5
6	AE	0000-0003	6	BACK PAIN	2009-04-27	2009-07-14	312	390	1	5
7	AE	0000-0003	7	SHINGLES	2009-06-09	2009-08-03	355	410	1	5
8	AE	0000-0003	8	ELEVATED CPK	2009-10-10	2009-12-18	478	547		

Table 4. Sample SDTM with date variables only

- Record 1 was accepted under Rule 7, which makes the conservative assumption to accept records with missing dates.
- Records 2–7 were accepted under Rule 5, because AESTDTC is earlier than the 2009-07-05 cutoff date.
- Record 8 was rejected because AESTDTC is after the cutoff date.

Notice in Table 4 how the reported end dates of two records fell after the cutoff date (thick border). This is a "problem" for standards compliance—no dates should be after the disposition date, etc. The authors contemplated rolling back end dates, but decided against doing so. End dates are discussed below.

DOMAIN WITH VISIT ONLY (TABLE 5)

It is not common in clinical data, but the standards allow domains to be constructed with visit information but no date. In this case, data cutting Rule 4 will be applicable.

One domain where visits are the sole timing information is TV (trial visits). Direct use of the %CUTDATA macro is thwarted by the absence of USUBJID.

Filtering the TV dataset for visits in the &VISITNUM_PLAN list would have accomplished the job, but would not be consistent with cutting of the other datasets.

This seemingly insurmountable problem was solved, however, without making an exception in the macro algorithm.

Prior to calling %CUTDATA, the reference visits dataset was searched for the subject ID who had the longest participation in the trial. This ID was then retained in the candidate TV domain. The %CUTDATA macro was now called, and all visits up to and including the cutoff visit got flagged as expected. Finally, USUBJID was dropped. Voila !

```
proc sql noprint ;
   select usubjid, max(dvdt) - min(dvdt)
   into :usubjid, :dur
   from &visit_dataset
   where dvdt is not null
   group by usubjid
   order by 2 desc ;
quit ;

%let usubjid = &usubjid ;

data work.tv ;
   set work.tv ;
   retain usubjid "&usubjid" ;
run ;

%CUTDATA (dsin         = work.tv ,
   dsout               = work.tvcut ,
   visitnum_plan       = &visitnum_plan ,
   visitnum_cutoff     = &visitnum_cut ,
   visitnum_next       = &visitnum_next ,
   visitnum_eos        = &visitnum_eos ,
   visit_dataset       = &visit_dataset ,
   action              = &cutdata_action ) ;
```

Now, even in the unlikely case that data cutting were to be performed when no subjects had reached the designated cutoff visit, the TV dataset would represent the sample population.

The results of this exercise are presented in Table 5.

Obs	DO–MAIN	VISIT–NUM	VISIT	VISIT–DY	TVSTRL	TVENRL	_FLG	_RULE
1	TV	1	Screening	-14	Signed Inform …	Met Inclusion …	1	4
2	TV	2	Day 1	1	Randomization	Completion of …	1	4
3	TV	3	Month 3	91	Start of …	Completion of …	1	4
4	TV	6	Month 6	182	Start of …	Completion of …	1	4
5	TV	9	Month 9	273	Start of …	Completion of …	1	4
6	TV	12	Month 12	365	Start of …	Completion of …	1	4
7	TV	18	Month 18	547	Start of …	Completion of …		
8	TV	24	Month 24	720	Start of …	Completion of …		
9	TV	30	Month 30	902	Start of …	Completion of …		
10	TV	99	End of Study	1085	Start of …	End of Partici …		

Table 5. Sample SDTM with visit but no date

DEMOGRAPHY DOMAIN (TABLE 6)

The %CUTDATA requirements were intentionally limited to date variables carrying the domain prefix. As such, reference start date (RFSTDTC) and reference end date (RFENDTC) will not be recognized. Should SDTM DM be run through the macro, all records would be accepted under Rule 1—when the domain has no timing variables, accept all records. This is not necessarily wrong, because study reports rarely discard study subjects.

Although inclusively is not compromised by overlooking reference start date, those subjects with an applicable cutoff date would have a reference end date RFENDTC inconsistent with other domains. See subjects 0001 and 0003 in Table 6. This may or may not be a problem when the cut SDTMs are used for analysis.

Obs	DO–MAIN	USUBJID	RFSTDTC	RFENDTC	Cutoff Dates
1	DM	0000-0001	2008-10-13	2011-05-29	2011-04-11
2	DM	0000-0002	2008-11-01	2009-06-23	Accept all records
3	DM	0000-0003	2008-06-20	2009-07-06	2009-07-05
4	DM	0000-0004	2008-07-30	2009-08-01	Accept all records

Table 6. Sample demography SDTM — without data cut

In the authors' studies, reference start and end dates were, respectively, the date of first and last dose, normally obtained by direct reading of raw exposure data.

In order to derive the reference dates consistent with the data cut, a faux exposure domain was prepared within the demography program. This faux domain consisted of USUBJID, exposure start date EXSTDTC, and exposure end date EXENDTC. Consecutive raw records with constant dose were not collapsed into dosing spans. Macro %CUTDATA was then allowed to operate on the faux domain and, from the result, RFSTDTC and RFENDTC were derived. Whether the %CUTDATA action parameter was set to flag or drop records, demography would be correct and consistent.

A WORD ABOUT END DATES

Some domains include start date (—STDTC) and end date (—ENDTC) to convey duration. When the data are cut, there is a real possibility that some end dates will fall beyond the cutoff date. Two adverse events records in Table 4 are this way. Logic considered by the authors to deal with this contingency is outlined in Table 7.

Domain Has	Compare cutoff date to —ENDTC and post-process
—STDTC —ENDTC —ENRF	• If the end date is after the cutoff date, make the end date missing and set —ENRF to "DURING/AFTER". • If the domain additionally has a study end day variable (–ENDY), make it missing too. • If the end date value is missing, or the end date is equal to or earlier than the cutoff date, do nothing.
—STDTC —ENDTC	• If the end date is after the cutoff date, roll back the end date to the cutoff date. • If the domain additionally has a study end day variable (—ENDY), roll it back too. • If the end date variable is missing, or the end date is equal to or earlier than the cutoff date, do nothing.

Table 7. Hypothetical Rules to Truncate End Dates

In the end, the authors decided against manipulating end dates. Think about it this way—data cutting is nothing more than a formalized "where clause". Imputing or blanking end date values is changing data. That's a statistical programmer's no-no.

Demography RFENDTC could be derived accurately by cutting the raw exposure data. Similarly, in the Exposure domain, source records were cut prior to rolling them up into constant–dosing intervals. As nothing in the analysis plan used other end dates, there was no reason to do anything.

CONCLUSIONS

Starting with the SDTM standard, requirements for a macro to cut datasets at an arbitrary visit were developed and implemented. The resulting macros were successfully used across all domains of several phase III protocols. Programmers seeking to automate routine operations are recommended to design their macros by reference to the SDTM standard, rather than particular protocols, for simplicity and generality. One size <u>can</u> fit all.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Anthony L. Feliu
Tel: (617) 665-4833
E-mail: ANTHONY.FELIU _AT_ GENZYME.COM

Genzyme, A Sanofi Company
500 West Kendall Street
Cambridge, MA 02142

Stephen W. Lyons
Tel: (617) 665-4830
E-mail: STEPHEN.LYONS _AT_ GENZYME.COM

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

APPENDIX

The production macro is reproduced below. To conform with page limitations, steps unrelated to the data cutting algorithm were omitted. Comment lines with leading asterisks show these abbreviations.

```
%macro cutdata(dsin   = ,
               dsout  = ,
               visitnum_plan   = ,
               visitnum_cutoff = ,
               visitnum_next   = ,
               visitnum_eos    = ,
               action          = ,
               visit_dataset   = ref.dvx,
               debug           = ) ;

***** Declare local macro variables ;
***** Change system options ;
***** Validate parameters ;
***** Exit when discrepancies found.  If all OK, document parameters in SAS log. ;

                                            /* Define attributes for new variables.    */
   %local attrib_cutdt attrib_rule attrib_flg ;
   %let attrib_cutdt = %str(_CUTDT length= 8  label='CUTDATA Cutoff Date' format = date9.) ;
   %let attrib_rule  = %str(_RULE  length= $5 label='CUTDATA acceptance rule') ;
   %let attrib_flg   = %str(_FLG   length= 8  label='1 = accept record, Null = reject') ;

                        /*-------------------------------------------------------------*/
                        /* ANALYZE THE INPUT DATASET.                                  */

***** Put domain name into macro variable DOMAIN. ;
***** Make macro variable booleans for each variable of interest (HAS_DTC, HAS_STDTC, etc. ;
***** Confirm DOMAIN and USUBJID exist, _RULE and _FLG do not exist. Exit if problem. ;

                        /*-------------------------------------------------------------*/
                        /* FIRST TIER LOGIC: DATASET LEVEL.                            */
                        /* If no timing variables, acccept all records in dataset.     */

   %if &has_dtc eq 0 and &has_stdtc eq 0 and &has_visitnum eq 0 %then %do ;

      data &dsout ;
         set &dsin ;

         %if %str(&action) eq %str(F) %then %do ;       /* ACTION=F, add flag variable. */
            attrib &attrib_rule &attrib_flg ;
            retain _rule '1' _flg 1 ;
        %end ;
      run ;

      %goto exit_cutdata ;
   %end ;

                        /*-------------------------------------------------------------*/
                        /* SECOND TIER: SUBJECT LEVEL. Find subject-specific dates.    */

   proc sort data = &visit_dataset (keep = usubjid visitnum visit dvdt) out  = work._dvx ;
      by usubjid visitnum dvdt ;
      where not missing(dvdt) ;
   run ;

   proc sql noprint ;                       /* In case subject skipped visits, must    */
      create table work._dvx_max as         /* search for any post-cutoff except EOS.  */
      select usubjid,
             8888 as visitnum,  'Latest date within accepted visits' as visit,
             max(dvdt) format = date9. as dvdt
      from work._dvx
      where visitnum in (%sysfunc(translate(&visitnum_plan, ",", " "))) and
            visitnum ne &visitnum_eos
      group by usubjid order by 1 ;
```

```
   create table work._dvx_post as
   select a.usubjid,
          9999 as visitnum,  'Earliest date beyond accepted visits' as visit,
          min(a.dvdt) format = date9. as dvdt
   from work._dvx a inner join work._dvx_max b
   on (a.usubjid = b.usubjid and
       a.dvdt > b.dvdt and
       a.visitnum ne &visitnum_eos)
   group by a.usubjid order by a.usubjid ;
quit ;

data work._dvx ;
   set work._dvx work._dvx_max work._dvx_post ;
   by usubjid visitnum dvdt ;
   if last.visitnum ;

   if visitnum = &visitnum_eos then visit = strip(visit) || ' (EOS)' ;
   else if visitnum = &visitnum_next then visit = strip(visit) || ' (NEXT)' ;
   else if visitnum = &visitnum_cutoff then visit = strip(visit) || ' (CUTOFF)' ;
run ;

           /* _DVX_TRANSPOSE has one record per subject with all visit dates.      */
           /* Decision is made there to accept all records or to fix cutoff date.  */

proc transpose data = work._dvx out = work._dvx_transpose (drop = _:) prefix = V_ ;
   format visitnum ;
   by usubjid ;
   id visitnum ;
   idlabel visit ;
   var dvdt ;
run ;

data work._cutdata ;                    /* WORK._CUTDATA is subject-level dataset. */
   set work._dvx_transpose ;            /* It has dates for the critical visits,   */
   attrib &attrib_rule &attrib_cutdt ;  /* and either cutoff date (_CUTDT), or     */
   _rule = '' ;                         /* rule under which all subject records    */
   _cutdt = . ;                         /* are accepted (S_RULE).                  */

                                        /* Make sure variables exist.              */
   if missing(v_&visitnum_cut) then v_&visitnum_cut = . ;
   if missing(v_&visitnum_next) then v_&visitnum_next = . ;
   if missing(v_&visitnum_eos) then v_&visitnum_eos = . ;
   format v_: date9. ;

                                        /* Can all records be accepted ??          */
   if (v_&visitnum_cut eq .) and        /* ... Yes, subject did not progress.      */
      (v_&visitnum_next eq .) then _rule = '2' ;

   else if (v_&visitnum_cut gt .) and   /* ...Yes, cutoff visit was last.          */
      (v_&visitnum_eos eq .) and (v_9999 eq .) then _rule = '3' ;

                                        /* .... Yes, cutoff visit date = EOS date. */
   else if ( nmiss(v_&visitnum_cut, v_&visitnum_eos) eq 0 ) and
      (v_&visitnum_cut eq v_&visitnum_eos) and (v_9999 eq .) then _rule = '3B' ;

                                        /* ... No, must determine a cutoff date.   */
   if _rule eq '' then do ;
      if (v_&visitnum_next gt v_8888) then _cutdt = v_&visitnum_next - 1 ;
      else if (v_&visitnum_next eq .) and (v_9999 ne .) then _cutdt = v_8888 + 7 ;
      else if (v_&visitnum_eos ne .) then _cutdt = v_&visitnum_eos - 1 ;
   end ;

   keep usubjid _cutdt _rule
        v_&visitnum_cut v_&visitnum_next v_&visitnum_eos v_8888 v_9999 ;

   rename _rule = S_RULE              v_&visitnum_cut  = V_CUTOFF
          v_&visitnum_next = V_NEXT   v_&visitnum_eos  = V_EOS
          v_8888 = V_ACCEPTMAX        v_9999           = V_POSTMIN ;

run ;

                      /*-------------------------------------------------------------*/
                      /* THIRD TIER: RECORD LEVEL. Merge WORK._CUTDATA with &DSIN.   */

proc sort data = &dsin out = &dsout ;  by usubjid ;  run ;
```

```
   data &dsout ;
      attrib &attrib_flg &attrib_rule ;
      merge work._cutdata (in = in_visits)  &dsout (in = in_data) ;
      by usubjid ;

      if in_data ;
      if not in_visits then put "Subject in &dsin not in &visit_dataset. " usubjid= ;

***** Derive numeric dates (--DT, --STDT) from SDTM variables (--DTC, --STDTC). ;

                        /* Apply subject-level decision from WORK._CUTDATA.  */
      if s_rule ne '' then do ;
         _flg = 1 ;
         _rule = s_rule ;
      end ;

                        /* RULE 4. Check visit number against list.     */
      %if &has_visitnum %then %do ;
         if (_flg ne 1) and (visitnum in (&visitnum_plan)) then do ;
            _flg = 1 ;
            _rule = '4' ;
         end ;
      %end ;

                        /* RULE 5. Accept --STDTC thru cutoff date.     */
      %if (&has_stdtc eq 1) and (&has_dtc eq 0) %then %do ;
         if (_flg ne 1) and  (. lt _&domain.stdt le _cutdt) then do ;
            _flg = 1 ;
            _rule = '5' ;
         end ;
      %end ;

                        /* RULE 6. --STDTC takes precedence over --DTC. */
      %else %if (&has_stdtc eq 1) and (&has_dtc eq 1) %then %do ;
         if (_flg ne 1) and (. lt coalesce(_&domain.stdt, _&domain.dt) le _cutdt) then do ;
            _flg = 1 ;
            _rule = '6B' ;
         end ;
      %end ;

                        /* RULE 6. Accept --DTC thru cutoff date.       */
      %else %if (&has_dtc eq 1) %then %do ;
         if (_flg ne 1) and (. lt _&domain.dt le _cutdt) then do ;
            _flg = 1 ;
            _rule = '6A' ;
         end ;
      %end ;

                        /* RULE 7. Accept when blank date unless domain has VISITNUM.  */
      %if (&has_stdtc or &has_dtc) and (&has_visitnum eq 0) %then %do ;
         if (_flg ne 1) and (nmiss(_&domain.dt, _&domain.stdt) eq 2) then do ;
            _flg = 1 ;
            _rule = '7' ;
         end ;
      %end ;

      %if %str(&debug) ne %str(ON) %then %do ; /* In DEBUG mode, keep date diagnostics.*/
         drop v_: s_rule _&domain.: _cutdt ;
      %end ;

      %if %str(&action) eq %str(D) %then %do ; /* ACTION=D, drop records not accepted. */
         if missing(_flg ) then delete ;
      %end ;

      %if (%str(&debug) ne %str(ON)) and (%str(&action) ne %str(F)) %then %do ;
         drop _flg _rule ;                     /* Drop flag variables when ACTION ne F */
      %end ;
   run ;

%exit_cutdata:
***** If validations failed, write explanatory messages to SAS log. ;
***** If macro ran normally, write confirmation message to SAS log. ;
***** Restore system options and perform other housekeeping. ;

%mend ;
```