PharmaSUG 2013 - Paper DS06

Designing and Tuning ADaM Datasets

Songhui ZHU, K&L Consulting Services, Fort Washington, PA

ABSTRACT

The developers/authors of CDISC ADaM Model and ADaM IG made enormous effort to give detailed guidance on implementing ADaM for clinical study data. However, due to the complexity of the clinical trials, they also give the users some flexibilities while implementing ADaM standards. Even so, in practice, creating CDISC-compliant ADaM datasets is not easy. In some cases, a bad choice in the early stage may result that the datasets are not CDISC-compliant and almost impossible to make them CDISC-compliant in the end. This paper will present author's practice on some critical choices while designing the data structures of ADaM datasets. The topics include: 1) whether to populate CRITy variables to each row or only the qualified rows, 2) whether to split datasets or categorize parameters in one dataset, 3) whether to utilize CRITy or add rows, 4) the relation between ANLzzFL and ABLFL/DTYPE, 5) mapping between AVAL and AVALC, 6) whether to dump to ADSL or create more datasets, and 7) whether to derive everything in an ADaM dataset based on SDTM data only or based on SDTM data and other ADaM datasets.

INTRODUCTION

When starting statistical programming for a study, we first need to think about designing ADaM datasets - how many analysis datasets to be developed and how to represent the data values. CDISC ADaM model and IG give guidance on the dataset and variable level metadata. However there are still a lot of situations where we need to make decisions to select the methods to implement the guidance. For example, in the ADaM IG V1.0, section 4.7, it gives two examples in populating CRIT - one populating for rows that met the selection criteria and another for all rows within a parameter. Also in IG V1.0, section 4.8.2, alternatives are given using analysis flag vs. creating multiple datasets to support analysis of the same type of data. Different methods suit for different situations. This paper, through examples, presents the thoughts and practices in designing ADaM data at certain situations when more than one way of implementation is available.

When designing ADaM datasets, one need make appropriate choices on the following seven issues. Making a right choice on each of the following issues will make the ADaM datasets serve one's analysis purposes better.

1. Populating CRITy variables to each row or only the qualified rows

According to ADaM IG, CRITy can be used using two different approaches: the first approach is to assign CRITy to each row of the same PARAM and set CRITyFL = Y for qualified rows; the second approach is to assign CRITy and CRITyFL for only qualified rows. Example 1.1 is to illustrate the first approach and Example 1.2 is to illustrate the second approach.

Example 1.1 Define CRITy for all records of the same parameter and set CRITyFL = Y for the records in which the criterion is met.

Table 1 Example **1.1** CRIT1 is populated for each row

USUBJID	VISIT	PARAM	AVAL	ANRHI	CRIT1	CRIT1FL
ABC-001	VISIT 2	AST (U/L)	180	34	AST>=5*ULN	Y
ABC-001	VISIT 2	ALT(U/L)	20	34	ALT>= 5*ULN	
ABC-001	VISIT 2	AST and ALT			AST and ALT >=5*ULN	
ABC-002	VISIT 2	AST (U/L)	178	34	AST>=5*ULN	Y
ABC-002	VISIT 2	ALT(U/L)	28	34	ALT>=5*ULN	
ABC-002	VISIT 2	AST and ALT			AST and ALT >=5*ULN	

In this example, variables CRIT1 are defined for each row no matter whether the criterion is met. CRIT1FL = Y if the criterion defined in CRIT1 is met.

Example 1.2 The second approach is that CRIT1 is populated only for rows in which the criterion is met. In addition, as in Example 1.1, for those records, CRIT1FL is set to 'Y'. This approach can be illustrated by the following example.

Table 2 Example **1.2** CRIT1 is populated only for rows in which the criterion is met

USUBJID	VISIT	PARAM	AVAL	ANRHI	CRIT1	CRIT1FL
ABC-001	VISIT 2	AST	180	34	AST>=5*ULN	Y
ABC-001	VISIT 2	ALT	20	34		
ABC-001	VISIT 2	AST and ALT				
ABC-002	VISIT 2	AST	178	34	AST>=5*ULN	Y
ABC-002	VISIT 2	ALT	28	34		

ABC-002	VISIT 2	AST and ALT				

As there is no any row met the criteria 3 “ASL and ALT great than 5 *ULN”, this criteria is not in the analysis data.

When coming to the table programming, it is not clear from the data derived in example 1.2 whether criteria 3 is evaluated or not. The most non-efficient part is that criteria 3 cannot be automatically feed into table program and need to be manually copied from either dataset program or variable metadata.

2. Splitting datasets or categorizing parameters

If is quite often that one need make a choice about whether to create separate datasets or put information within one single dataset in which variables PARAM and/or PARCATy are used to group parameters. For example, in oncology studies, tumor response is often assessed by an investigator and by a central judicator in parallel; The two sets of parallel assessments are both analyzed. In this case, one has two choices: the first one to put both investigator's and the adjudicator's assessments in the same ADaM dataset; the second choice is to put the investigator's results in one ADaM dataset and the adjudicator's assessment results in another dataset.

Example 2.1 Use PARAM and/or PARCATy to group investigator's assessments and adjudicator's assessments. In this implementation, PARAM has two different values for two different evaluators. See Table 2.1 for details. It is worth pointing out that this implementation complies with ADaM standards.

Table 2.1.1 Using different PARAM for the records from different source

USUBJID	PARCAT1	PARM	PARAMCD
ABC-001	Response Assessment	Response from investigator	RESPINV
ABC-001	Response Assessment	Response from Central Judicator	RESPCNT

However, it is not rare to see implementations in which PARCATy rather than PARAM is used to group assessments illustrated in Table 2.1.2.

Table 2.1.2 INCORRECT way to categorize PARAM and PARAMCD

USUBJID	PARCAT1	PARM	PARAMCD
ABC-001	Investigator Response Assessment	Response Assessment	RESPONSE

ABC-001	Judicator Response Assessment	Response Assessment	RESPONSE

In the implementation shown in Table 2.1.2, PARAM is identical for both investigator's and adjudicator's assessments while PARCAT1 is different. Are both implementations CDISC compliant? The ADaM analysis parameter (PARAM) has to contain all of the information needed to uniquely identify a group of related analysis values [1] and PARCATy is a categorization of PARAM [2]. Therefore, a PARCATy can contain more than one PARAM and PARAMCD. But a PARAM can not belong to more than one PARCATy. In implementation shown in Table 2.1.2, PARAM Response Assessment belongs to two different PARCAT1 values, which contradicts CDISC standards.

Example 2.2 Splitting the response assessments into two ADaM datasets.

An alternative solution to Example 2.1.1 is to put the response assessments into two separate datasets: one for investigator's assessments and the other for adjudicator's assessments. This implementation is shown in the following table.

Table 2.2 Splitting the records from different sources into two datasets

Dataset	**USUBJID**	**PARM**	**PARAMCD**
ADASSINV	ABC-001	Response	RESPONSE

Dataset	**USUBJID**	**PARM**	**PARAMCD**
ADASSCNT	ABC-001	Response	RESPONSE

In this implementation, PARAM is the same for both investigator's and adjudicator's assessments (Response). Consequently, PARCAT1 can be identical in two datasets.

Whether to split the dataset or use variables to group parameters, one need consider the pros and cons of the two approaches and the analysis needs of the studies. The advantages and disadvantages of two approaches can be summarized as follows:

Table 2.3 Advantages and disadvantages of using PARAM and splitting dataset

Approach	**Pros**	**Cons**	**When to use**
#1. Using different PARAM's for the records from different sources	Easy to compare the differences for records from	Complicated dataset programming	Derivation rules for data from different sources are similar; Assessment schedule are similar

	different sources; Less datasets,		
#2. Splitting the records from different sources into 2 dataset	Same PARAM and PARAMCD; Table programming is simpler	More datasets, more resource needed for development, validation and maintenance	Derivation rules are not similar; Different assessment schedule

3. Utilizing CRITy or adding a row for values in a single parameter

Variable CRITy in BDS datasets is used to identify a pre-specified criterion. Can the criterion be anything else needed for analysis?

Example 3.1 In some studies the smallest value of a subject's post baseline heart beat (HR) measurements in ADVS is needed. Can it be flagged using CRIT1? No. The CRITy cannot be used for multiple rows criteria even for a single parameter as it is explained in ADaM IG section 4.7.1, "If the definition of a criterion uses values located on multiple rows (different parameters or multiple rows for a single parameter), then a new row must be added". At this case a row is added with the smallest post baseline value and DTYPE set as worst value carried forward WOCF or customized as minimum value carried forward MINOCF.

Table 3.1 Adding a row for multiple row criterion within a single parameter

USUBJID	PARAMCD	AVAL	DTYPE
ABC-001	HR	50	
ABC-001	HR	67	
ABC-001	HR	61	
ABC-001	HR	50	WOCF

Example 3.2 At the same situation, ANLzzFL can be used to identify the row for analysis.

The variable ANLzzFL is designed to (together with other variables) specify the rows that fulfill specific requirements for one or more analyses. If the minimum value of HR is needed for analysis, ANLxxFL can be utilized to flag the row for the analysis. In this way, no additional row is needed.

Table 3.2 Using ANL01FL to identify the row for analysis

USUBJID	PARAMCD	AVAL	ANL01FL
ABC-001	HR	50	Y
ABC-001	HR	67	
ABC-001	HR	61	

As shown in Table 3.2, using ANL01FL has the advantage of reducing the number of observations. But using variable CRITy is clearer since the value of CRITy is self-explaining while ANL01FL is not.

4. Relation between ABLFL/DTYPE and ANLzzFL

Consider the case where the baseline value is the average of values at Screening visit and Visit 2. At the same time, by-visit analysis is also needed. How to flag ABLFL and ANLzzFL so that table programming is easier?

In this case, ANLzzFL can be used to flag the records that are to be used for by-visit analysis, while ABLFL is used to identify baseline records. ANLzzFL and ABLFL are two independent variables. That is, records with ABLFL = Y may or may not have ANLzzFL = Y. ADaM IG does not require that all baseline records have ANLzzFL = Y. This implementation is shown in Table 4.1. In this example, subject ABC-001 has two pre-treatment visits and, therefore, an extra record is created for baseline for which ANL01FL is not Y but ABLFL = Y. Subject ABC-002 has only one pre-treatment visit and that record is flagged as baseline for which ANL01FL = Y. In addition, subject ABC-002 has two duplicate records at Visit 3. An extra record is created as the average of the two duplicate records. The average is to be used for by-visit analysis. ANL01FL is set to Y for this average record. Further, through this example, one can see that ABLFL and ANLzzFL are not related. Further, subject ABC-001 has ANL01FL = blank when DTYPE = AVERAGE; However, subject ABC-002 has ANL01FL = Y when DTYPE = AVERAGE. Therefore, DTYPE and ANLzzFL are not related, either.

Table 4.1 Records with ABLFL = 'Y' and ANL01FL blank

USUBJID	PARAMCD	AVISIT	AVAL	ABLFL	ANL01FL	DTYPE
ABC-001	HEARBT	SCREENING	70		Y	
ABC-001	HEARBT	VISIT 2	74		Y	
ABC-001	HEARBT	BASELINE	72	Y		AVERAGE
ABC-001	HEARBT	VISIT 3	75		Y	
ABC-002	HEARBT	VISIT 2	80	Y	Y	
ABC-002	HEARBT	VISIT 3	78			
ABC-002	HEARBT	VISIT 3	76			
ABC-002	HEARBT	VISIT 3	77		Y	AVERAGE
ABC-002	HEARBT	VISIT 4	82		Y	

5. Mapping AVALC to AVAL

AVALC is character analysis value and it is an one-to-one mapping to AVAL. People usually map numeric value in SDTM to AVAL and character value in SDTM to AVALC. Generally, it has no problem. However, mapping AVALC to AVAL can be troublesome when there are imputation rules for AVAL.

Example 5.1 For all calcium values in lab data, if there are values with character version of '<=' or '>=', the numeric version of the value is missing. When the imputation rule 'striping off > or < sign to get numeric value' is directly implemented, the following mapping between AVALC and AVAL will appear.

Table 5.1.1 INCORRECT mapping between AVAL and AVALC

PARAMCD	LBSTRESC	AVALC	AVAL
CALCIUM	2.5	2.5	2.5
CALCIUM	<2.5	<2.5	2.5
CALCIUM	>=2.5	>=2.5	2.5

In this implementation, three different character values ('2.5', '<= 2.5', and '>= 2.5') are all mapped to numeric value 2.5, which is not one-to-one mapping. In this case, the implementation can be modified as follows: derive AVAL according to the imputation rule, leave AVALC blank. To get original values, keep the original variable LBSTRESC in the analysis dataset. This implementation can be illustrated as in

Table 5.1.2 Alternative way to map AVAL

PARAMCD	LBSTRESC	AVALC	AVAL
CALCIUM	2.5		2.5
CALCIUM	<2.5		2.5
CALCIUM	>=2.5		2.5

For parameter CALCIUM, the one-to-one mapping rule is not applicable and not violated since all values of AVALC are not populated.

6. Dumping to ADSL or creating separate datasets

It is quite often that one needs to decide whether to put all subject-level information into ADSL or in ADSL and separate datasets. Subject-level information could include demographic data, prior treatment/disease information, discontinuation information, and drug exposure information. In general, making decision on this issue need take into account the complexity of the studies and the analysis needs

Example 6.1. Creating discontinuation analysis dataset or put discontinuation information in ADSL?
Discontinuation dates and reasons are important. When designing datasets, one needs to decide whether to create an analysis dataset to save discontinuation information or put discontinuation information in ADSL
In general, discontinuation information such as the date of and reason for discontinuation from treatment/study can be in ADSL when there is only one or two periods and the number of reasons are predictable. However, a separate dataset is more appropriate to hold discontinuation reasons or other information when there are more than one or two periods or the number of discontinuation reasons are not predictable. Important discontinuation dates usually need to be in ADSL so that they can be used to calculate time to event or determine treatment emergent adverse events. Also, death date and reason for death usually need be put in ADSL to derive other important dates or analysis such as overall survival analysis.

Example 6.2. Creating exposure analysis dataset or put exposure information in ADSL?
If there are a lot of analyses for exposure data such as exposure duration, average daily dose by period, and interruption due to AE by period, it is better to put exposure information in a separate analysis dataset. The important drug, period start and stop dates will always be in ADSL.

7. Deriving an ADaM dataset based on SDTM data only or based on SDTM and other ADaM datasets

As required by regulatory agencies, under CDSIC umbrella, the appropriate path of deriving analysis datasets is from raw data to SDTM and then from SDTM to ADaM. Can we derive ADaM from another ADaM dataset?
Example 7.1 Treatment emergent life-threatening adverse events are identified and imputation rules for missing dates/codes are implemented in analysis dataset ADAE. For time to treatment emergent life-

threatening AE analysis and other time to event analyses, another analysis dataset ADTTE is needed. Do we have to re-derive treatment emergent flag, or impute missing start dates from data in SDTM or can we get the information derived in ADAE? As ADaM or ADaM IG has no restrictions on using another ADaM dataset, the information needed in ADTTE can be from ADAE.

CONCLUSIONS

This paper presented the author's thoughts and practice on seven important issues about implementing CDISC-compliant ADaM datasets. Making good choices on these issues can help make the datasets CDISC-compliant and serve the analysis purpose well.

REFERENCES

[1] CDISC Analysis Data Model ,Version 2.1.

[2] CDISC ADaM Implementation Guide, Version 1.0.

[3] CDISC SDTM Implementation Guide, Version 3.1.2.

ACKNOWLEDGEMENTS

Songhui Zhu thanks K&L Consulting Services for the support he received.

CONTACT INFORMATION

Songhui Zhu, Associate Director

K&L Consulting Services,

Fort Washington, PA

songhui.zhu@klserv.com