

Streamlining ADaM Dataset Specifications

Hunter Vega, CROS NT LLC

ABSTRACT

In a large group of statistical programmers, you often find that people have a varied style for how they might create an analysis dataset specification. It could either be too detailed so that it compromises the independent programming concept or lacking in key information, e.g. how flags are to be derived. In an effort to consolidate some core practices for analysis dataset programming, we put together a training for our statistical programming and biostatistics teams to go over basic ADaM dataset concepts and the process to be followed at CROS NT. The purpose of the training was to create guidelines for how to create an analysis dataset specification, how much information one should be expected to include and when the biostatisticians should review. This collaboration of our biostatistics and statistical programming groups has improved the quality of our analysis dataset specifications as well as improved the efficiency of our programming of these datasets.

INTRODUCTION

Many CROs often employ statistical programmers from varied backgrounds and experience levels. As there is no perfect way to write an analysis dataset specification, we found that this is a task that gives varied results depending on the original specification writer. In an effort to improve the quality of our analysis datasets specifications, we designed a training to give our statistical programming and biostatisticians some general guidelines for specification writing. The training was presented as a two part series, as described below. The first part of the training was an overview of key objectives to remember when writing an analysis dataset specification as well as a review of the appropriate ADaM dataset types to be used. The second half of the training involved a discussion of two examples of analysis dataset specifications and programming tasks that we asked the statistical programming and biostatistician team members to practice with in order to highlight issues that often happen in a real study.

HOW TO CREATE A SPECIFICATION

Analysis dataset specifications almost always include derivations needed for a statistical analysis. This means that the study statisticians need to be involved in the process of specification-writing from the beginning. We encourage study statisticians to review any derivation described in the specifications and also just to be available as references for questions from statistical programmers. It is also important that the statistical programmers are following the rules set out by the CDISC documentation so that the ADaM datasets will be CDISC compliant for submission.

WHAT ARE THE MAIN GOALS OF ADAM DATASET SPECIFICATIONS?

- To give clear instructions for data values to be included in the ADaM datasets and also to provide traceability with respect to the source data
- Include any derivations, imputations, or references to the statistical analysis plan (or SAP) as needed to provide rules for how the analysis values are created
- Follow the rules laid out in CDISC ADaM Implementation Guides for variable names, labels, and the types of data that correspond to different styles of ADaM datasets (subject level ADSL, BDS, and OCCDS)
- Include all derivations needed for the statistical analysis. This ensures that no major data manipulation is programmed in the individual programs for the statistical outputs (tables, listings, or

figures). This also allows all derived values to be more easily reviewed by the study statistician

GOAL 1 - TRACEABILITY

If a value originates directly off the SDTM or raw data, very little information is needed in the specification as shown in Table 1 below.

Domain	Description of Domain	Variable Name	Label Name	Variable Type	Derivation Algorithm
ADLB	Laboratory Test Results Analysis Dataset	PARAM	Parameter	Character	LB.LBTEST concatenated with LB.LBSTRESU
ADLB	Laboratory Test Results Analysis Dataset	PARAMCD	Parameter Code	Character	LB.LBTESTCD

Table 1. ADLB example specification

Oftentimes a core set of variables from the ADL is kept on all other analysis datasets, such as variables with subject ID information, population flags, and treatment information. In order to reference these values that are directly kept from the ADL, you might mention the source dataset in the derivation algorithm as simply "ADL" or "ADL.FASFL".

GOAL 2 - CLEAR DERIVATIONS

If a value is derived, then the derivation must be described in a non-technical manner (no SAS code if possible) so that a person who is not a programmer can also understand the logic of the derivation. Alternatively, if the derivation is described in the SAP, then it is also appropriate to simply reference the corresponding SAP section for the derivation of that specific value. The idea is to give clear instructions for the derivation while also allowing some flexibility for the programmer validating the dataset to program the derivation a different way than the original programmer. This allows for a higher quality ADaM dataset by creating a greater chance of finding any issues in the data or with how the original analysis dataset is programmed. In the example below, the specification's author gives a description in layman's terms that also references the study SAP.

Domain	Description of Domain	Variable Name	Label Name	Variable Type	Derivation Algorithm
ADEF	Efficacy Analysis Dataset	TMFRM	Timeframe	Character	Assign timeframe value based on ADY (Analysis Relative Day) and time frame windows described in SAP section 6.1.

Table 2. ADEF example specification

In the training, examples were also given of derivations that might be considered too specific or too similar to SAS code. For example, if you wanted to derive a mean VAS score across 4 different visits, you could use the derivation as shown here.

Domain	Description of Domain	Variable Name	Label Name	Variable Type	Derivation Algorithm
ADEF	Efficacy Analysis Dataset	MEANVAS	Mean VAS Score	Numeric	(VASV1+VASV2+VASV3+VASV4)/4

Table 3. ADEF example specification

If one of the VASV1-VASV4 scores is missing, then this derivation might produce a missing value when using SAS. Alternatively, defining the derivation as “mean(VAS1,VAS2, VAS3, VAS4)” which references the mean() function in SAS, would result in a mean value being calculated even if all 4 scores were not available. A better approach would be to give the description as “derive the mean of the available VAS scores: VASV1, VASV2, VASV3, and VASV4.” It would be a more thorough approach to also mention how to handle missing values. For example, you could add this to the derivation description previously given: “If more than one VAS score is missing then the MEANVAS value should not be derived per SAP section 4.2”.

GOAL 3 - FOLLOW CDISC GUIDELINES

The CDISC group maintains a detailed set of documentation rules for ADaM datasets. The most recent versions of the Implementation Guide as well as the Analysis Data Model documents should always be referenced when creating ADaM datasets. Although there is some flexibility in ADaM datasets, such that variables can be added that are not specifically mentioned in the documentation, it is important to make sure data types are formatted in the correct style as given in the ADaM documentation. There are three main types of ADaM datasets: ADSL, BDS, and OCCDS. Here is an overview of these types of datasets:

ADSL – Subject Level Analysis Dataset

Every study should have an ADSL dataset that includes one record for each subject in the study. This dataset should include population flags, treatment information, demographic information, and any general subject level information.

BDS – ADaM Basic Data Structure Analysis Dataset

This standard data layout can be used for many types of “findings” data such as: lab values, vital signs, ECG results, efficacy, etc. The dataset may contain one or more records per subject, per analysis parameter, and per time point. Only records with available data should be kept on this dataset. For example, if a subject does not have any laboratory results information in the database, they should not be included in the ADSL.

Specific variables are required to be considered a BDS-style dataset: PARAM, PARAMCD, and AVAL/AVALC. You do not need to keep both AVAL and AVALC if they are not different values, but at least one must be used. The PARAM value should uniquely describe what is in the AVAL/AVALC variable. If applicable it should include units, type, location, or position (but not timing information). PARAM and PARAMCD should also have a 1 to 1 correspondence.

Another variable of interest often used in BDS style datasets is the ANLZZFL flag. This analysis flag is used to identify records to be used in analysis. For example, if records are windowed to an analysis visit (AVISIT) value and a subject has more than one record assigned to the same visit, then ANLZZFL could be used to identify which record should be summarized in the analysis outputs.

OCCDS – Occurrence Data Analysis Dataset

These datasets are the style used for “event” or “intervention” type data. Examples of data that fit this style are adverse events, concomitant medications, and medical history. Occurrence data often includes a structured hierarchy of dictionary coding categories. This dataset style does not require the use of PARAM or AVAL, but they generally must include a XXTERM variable or a CMTRT variable (in the case of a concomitant medications dataset).

GOAL 4 - INCORPORATING STUDY STATISTICIAN INPUT

Study statisticians are responsible for the content of statistical analysis output, so it makes sense that they be involved with the derivation of any values needed for the statistical analysis. Our training emphasized the role of the study statistician in producing ADaM dataset specifications and how they need to be involved in the analysis data specification process at the time of analysis dataset creation, rather

than at the statistical review of any statistical summary output. We suggest a process for ADaM dataset creation from start to finish that looks something like this:

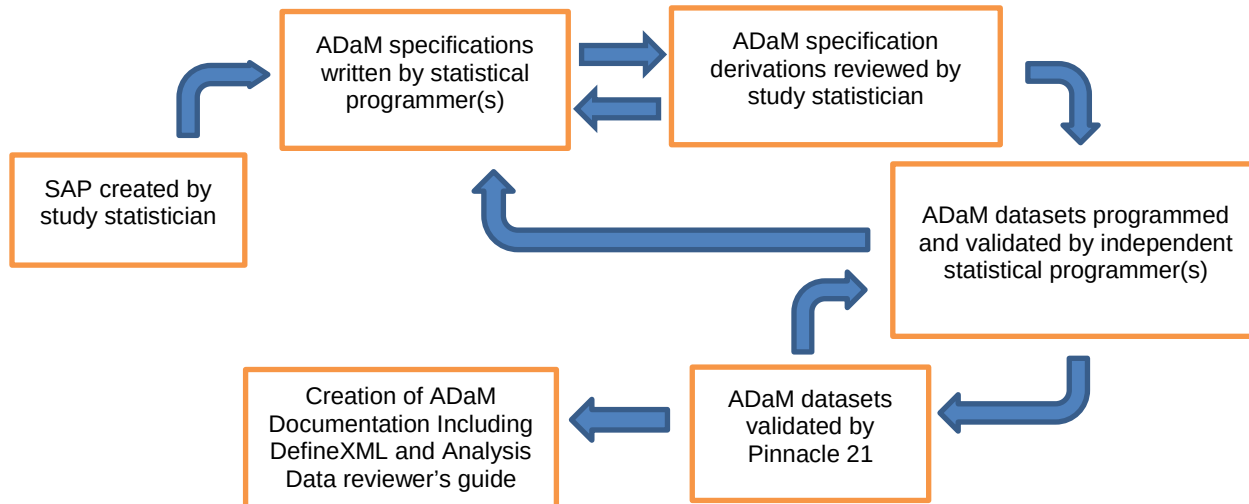


Figure 1. Flow of ADaM dataset and documentation creation

As shown above, oftentimes an ADaM dataset specification will need adjustment after comments are given by the study statistician or after the actual dataset programming begins. It is important that the dataset specifications reflect any changes that might occur because of original programming or validation findings. It is also true that changes may occur in the ADaM dataset specifications and programming because of findings arising during the validation of ADaM datasets using Pinnacle 21© software. The process flow provided above shows that any changes to derivations in the specifications that occur after the original study statistician review should be reviewed again once all the specification updates are finalized.

HANDS ON PRACTICE

When presenting the training, it was important to the programming team to make sure that these concepts were clearly understood and could be implemented in a real study situation by both the statistical programmers and the statisticians. With that thought in mind, two examples were given to the statistical programming and statistician group to test their intake of the ADaM specification process guidelines.

ADAM SPECIFICATION REVIEW BY VALIDATION PROGRAMMERS

In the first example, programmers validated an ADAE dataset containing adverse event data based on an ADaM dataset specification created by the trainers. The specification contained intentional errors with the purpose of encouraging the statistical programmers to think critically about any ADaM specification they are given and to question any derivation that they think is incorrect. Statistical programmers need to look at both the specification and the study documents, such as the SAP and the output specifications for the tables, listings, and figures, to make sure that the derivations listed in the ADaM specification are accurate. For example, the specification contained a derivation for AENDY (Relative End Day of Adverse Event), but the derivation description referred to the start date of the adverse event instead of the end date of the adverse event. There were also multiple comments from the statistical programmers about the derivation for the treatment emergent flag. Some responded that the derivation description was not clear, but the derivation algorithm was copied from a SAP from a real study. This is a case where a SAP description of a derivation might not be clear enough to use in a specification and therefore it is helpful to add any details that might clarify the description of the logic needed to derive the value in question. This

is also a case where communication with the study statistician is essential to clarify any concept that is not clear from the SAP text.

CREATE YOUR OWN ADAM SPECIFICATION

The second example asked the statistical programmers to create an analysis dataset for vital signs data (ADVS). Mock table shells and endpoints of interest that needed derivation were provided as the source documentation for the statistical analysis. The study statisticians were also grouped with statistical programmers to review their specifications and provide feedback in the same manner they would for a real study.

The mock tables provided to programmers and statisticians show a univariate summary of observed results and change from baseline results. In addition, another example table displays endpoints of interest that need to be derived:

Table 2.1
Number (%) of Subjects with Heart Rate < 60 bpm / Safety Population

Timepoint	Treatment A			Repeat for Treatment B, C, D		
	n	N	%	n	N	%
Baseline						
2 Weeks						
3 Months						
6 Months						
12 Months						
24 Months						

Repeat table 2.1 style for:

Table 2.2 Number (%) of Subjects with SBP > 140 mmHg / Safety Population

Table 2.3 Number (%) of Subjects with Percent Change from Baseline in Heart Rate > 5% / Safety Population

Display 1. Example Vital signs mock table

Statistical programming findings and study statistician findings were both discussed in the training. There were several issues found with statistical programmers' specifications:

- BASE value defined, but the ABLFL flag was not included. Per the ADaM Implementation guide, if you are keeping baseline values in BASE then it is also required that a ABLFL flag is used to identify the baseline record for that subject/parameter
- Analysis result needs to be kept in AVAL or AVALC. As specified in the original training, the AVAL variable is required in order for the ADaM dataset to be classified as a BDS style dataset
- CRIT# flag used but not CRIT#FL or CRIT#FN. If a CRIT# is used in an analysis dataset, then the CRIT#FL/CRIT#FN flag must also be included. The CRIT# flag will contain the description of the criteria being identified, while the CRIT#FL will equal Y or Y/N to indicate if the data record meets the criteria of interest
- Keep derivations simple. Some derivation text provided was given as SAS code which is too technical for a dataset specification.

We also reviewed the comments statisticians gave for the statistical programmers' specifications:

- For the CRIT1/CRIT1FL flag: be careful on the definition of AVAL<60. This could include missing AVAL values. This is very likely to occur in a real study, this comment from the statistician would help the programmer have a more precise derivation here.
- PCHG definition of (AVAL-BASE)/BASE *100 for any AVISITN>1 needs to add where BASE^=0 to the derivation description. This comment would help the programmers avoid any division by zero

warnings in the program log.

- The temperature value needed to be converted from Fahrenheit to Celsius for the analysis. The source documents provided that the source data for temperature was given in Fahrenheit units, but the tables wanted the temperature summarized with the Celsius units. Multiple programmers missed this detail in the source documents. This could have been a major issue if it was not caught until the final statistical review of the output before delivery to a sponsor.

CONCLUSION

The goal of this training series was to encourage the cooperation of the statistical programming group and the statistician group to develop a smarter process for ADaM dataset specification creation and review. Guidance for CDISC rules and derivation examples were given to provide statistical programmers a template for better quality dataset specifications. These specifications can be easily understood by other statistical programmers and statisticians, as well as any regulatory reviewer who might look at the derivation descriptions in the DefineXML. The real world example portion of the training especially encouraged the cooperation of the statisticians earlier in the review process as well. Reviewing the findings from both the statistical programmers and the statisticians highlighted the types of findings one might find in a real study situation. Our hope is that this training will improve our ADaM dataset specifications as we move forward using this improved ADaM specification creation and review process.

REFERENCES

CDISC, "Analysis Data Model Implementation Guide" - Version 1.1, issued on 2016-02-12.

CDISC, "Analysis Data Model (ADaM)" - Version 2.1, issued on 2009-12-17.

CDISC, "ADaM Structure for Occurrence Data (OCCDS)" - Version 1.0, issued on 2016-02-12.

ACKNOWLEDGMENTS

I would like to thank Laura Edwards and Silvia Faini for their review and suggestions.

CONTACT INFORMATION

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

Hunter Vega
CROS NT, LLC
(919) 929-5015
hunter.vega@crosnt.com
www.crosnt.com