PharmaSUG 2015 - Paper DS11

It Depends On Your Analysis Need

Sandra Minjoe, Accenture Life Sciences, San Francisco, USA

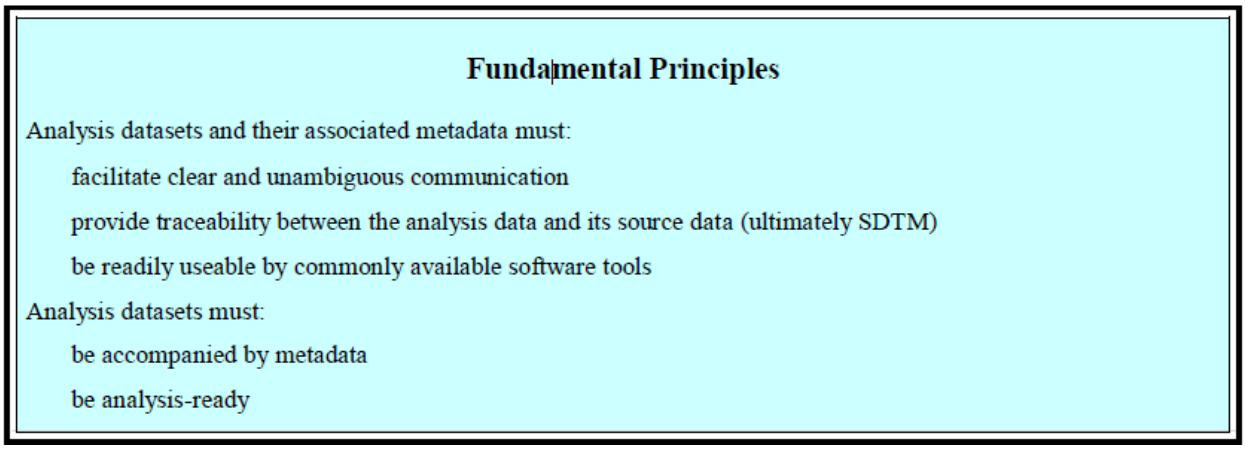
ABSTRACT

As an ADaM consultant, I'm regularly asked by friends and coworkers fairly general, and what they think are "quick" questions. My usual response to these types of questions is often "It depends on your analysis need," which I suspect is a bit disappointing to hear. However, my vague answer is because the question itself doesn't have enough detail for me to reply with exactly one correct or best answer. In this paper I delve into some of these questions that I've been asked and describe some of the different answers that each has … depending on your analysis need.

INTRODUCTION

The fundamental principles of ADaM are shown in this box, copied from Section 3.1 of the CDISC Analysis Data Model v2.1 document (ADaM 2.1)[1]:

Figure 1: ADaM Fundamental Principles



Fundamental Principles

Analysis datasets and their associated metadata must:

- facilitate clear and unambiguous communication
- provide traceability between the analysis data and its source data (ultimately SDTM)
- be readily useable by commonly available software tools

Analysis datasets must:

- be accompanied by metadata
- be analysis-ready

Notice that these fundamental principles focus on analysis need and understanding, not dataset structure. Data structures are mentioned later on in that document, within the ADaM Implementation Guide v1.0 (ADaMIG 1.0)[1], and within other ADaM documents.

The Computational Science Symposium (CSS) working group titled "Optimizing the Use of Data Standards" has developed an Analysis Data Reviewer's Guide (ADRG)[2] template, instructions for use, and examples. This reviewer's guide was developed with input from the ADaM team and is designed to be submitted along with the study data and define to provide additional information about the study and analysis performed.

I've been part of the CDISC ADaM team since 2001, and the CSS Optimizing the Use of Data Standards working group since its inception in 2012. I currently work as an ADaM Consultant, helping others apply the ADaM and CSS concepts for their particular study needs. I can't speak for the ADaM or CSS teams, but my understanding of ADaM and the ADRG, and thus advice for others, is:

- The ADaM fundamental principles take precedence over any decision around ADaM structure. First and foremost, an ADaM dataset must follow the ADaM fundamental principles. When it makes sense for analysis needs, then we design the dataset to fit one of the ADaM structures.
- Although not part of ADaM, the ADRG is mentioned in the FDA Technical Conformance Guide and Catalog[3], and should be part of any submission that includes ADaM.

In the sections below, I walk through a set of possible answers for some specific questions I've received, regarding datasets and variables:

1. Should I keep all the lab tests from SDTM LB in an ADLB analysis dataset?
2. Do all baseline characteristics need to be derived in ADSL?

regarding structure:

3. Does my ADEX or ADPK dataset need to use the BDS structure?
4. Can I use the ADAE structure for concomitant medications or medical history?

and regarding metadata:

5. Do I need to include parameter or results metadata?
6. Do I need to include an ADRG?

This paper contains opinions of the author and doesn't represent any group.

CATEGORY: DATASETS AND VARIABLES

1. SHOULD I KEEP ALL THE LAB TESTS FROM SDTM LB IN AN ADLB ANALYSIS DATASET?

Answer: It depends on your analysis need.

There are many factors that can affect whether all lab tests belong in the analysis dataset. Is every test analyzed on a table, or are some tests not used in analysis? Will listings be generated from the analysis dataset, including those that need to contain lab tests that are not otherwise analyzed in a table? Are there so many subjects and visits that it might make analysis and review easier to have more than one analysis dataset? Are there other records that might not make sense to include in an analysis dataset?

It depends on tests used/not used in analysis

Many of the lab tests that are collected never seem to end up on a table. For example, the analysis tables might include only a few tests from a chemistry panel, even though the laboratory will generally perform every test in their panel.

ADaM 2.1[1] states "It is not required that the data be collated into analysis-ready datasets solely to support data listings or other non-analytical displays, although some may choose to do so." This means you may choose to not include in ADLB the lab tests that aren't used in analysis, though ADaM doesn't disallow the practice of including non-analyzed lab tests in ADLB.

According to ADaMIG 1.0[1], "The ADaM methodology is to include all observed and derived rows for a given analysis parameter", even those rows that are not used in analysis. For example, if there are two possible week 4 records and you choose to use only one of them in your analysis, including both of these records makes the analysis dataset more robust. Sensitivity analysis wouldn't be possible if you brought over from SDTM only the subset of rows that will be analyzed for each lab test. To help eliminate confusion with these multiple rows, analysis flags (ANLzzFL) can be used to show which rows within a parameter were used in a particular analysis.

If you have many lab tests that are collected and not analyzed, you might want to base your decision to keep or not keep them on whether you'll use the dataset for generating listings, the size of the analysis dataset, and other factors described below.

It depends on whether you'll use the dataset for generating listings

Listings are produced for a variety of different reasons, for example to a) review all the data that was collected or b) to show all the records that contributed to a specific table. Listings can be generated from SDTM, ADaM, or even other data. In some cases, SDTM data merged with a few ADSL variables can be very useful for listings.

When ADaM data is used for listings, it would be beneficial to include in the ADaM dataset all the tests needed for the listings. As stated above and in the ADaM 2.1[1] document, this is certainly allowed.

It depends on the size of your dataset

Clinical trial datasets, especially with laboratory data, can get very large. SDTM provides a way to split large datasets, such as LB and QS, into smaller ones for ease of use. ADaM instead simply suggests creating the "optimum" number of analysis datasets. In ADaM, we can create multiple analysis datasets from one SDTM input data – for example, splitting LB by panel into ADLBCHEM, ADLBHEM, ADLBURIN, etc. In addition to splitting, analysis datasets can be made smaller by excluding all the lab tests that will not be analyzed. Be careful not to split or remove too much, because you still need the dataset to be "analysis ready".

It depends on the type of test

Many companies make use of an SDTM feature to denote when a whole group of tests was not done. From the SDTM Implementation Guide version 3.2 (SDTMIG 3.2)[4]:

Figure 2: SDTM Tests Not Done

If a group of tests were not done:
• --TESTCD should be --ALL
• --TEST should be <Name of the Module>
• --CAT should be <Name of Group of Tests>
• --ORRES should be null
• --STAT should be "NOT DONE"
• --REASND, if collected, might be "Specimen lost"

For example, if urinalysis is not done then:
• LBTESTCD should be "LBALL"
• LBTEST should be "Labs Data"
• LBCAT should be "URINALYSIS"
• LBORRES should be NULL
• LBSTAT should be "NOT DONE"
• LBREASND, if collected, might be "Subject could not void"

The lab test code of "LBALL" will likely never be needed in analysis, nor even in listings.

This is just one example of an SDTM "test" that may not translate to an ADaM parameter. If an SDTM test won't be useful for any analysis, there is no reason to create a corresponding parameter in ADaM.

It depends on your analysis needs

As described above, there is no one right answer for all cases. In each situation, the analysis need drives the decision on which lab tests from SDTM to keep in a corresponding ADaM dataset.

2. DO ALL BASELINE CHARACTERISTICS NEED TO BE DERIVED IN ADSL?

Answer: It depends on your analysis need.

There are a few factors that can affect whether all baseline characteristics belong in the ADSL dataset. Are all baseline characteristics applicable to the majority of analyses, or are some used only for a small subset of analyses, such as efficacy? Does the baseline derivation in ADSL need to match the baseline derivation in a BDS dataset? Do we have complicated baseline derivations that would make traceability difficult in a one-record-per-subject ADSL structure?

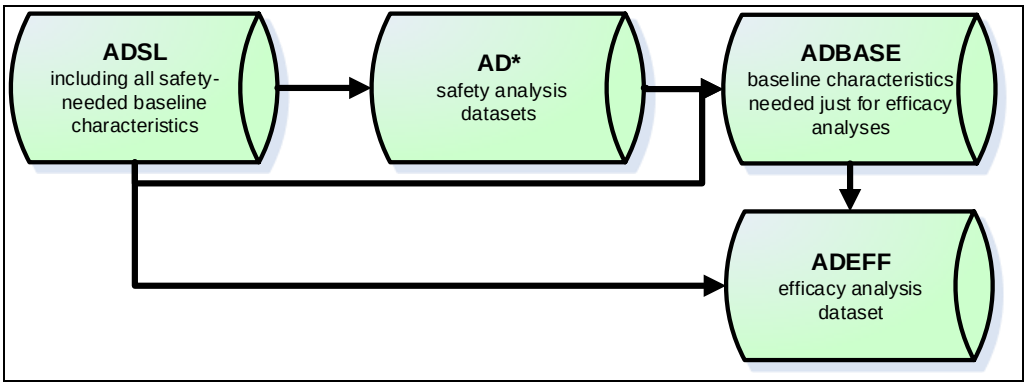
It depends on whether the baseline characteristics are needed across all/most analysis datasets.

Neither ADaM 2.1[1] nor the ADaMIG 1.0[1] specifically state that all baseline characteristics are required to be in ADSL. The USFDA Technical Conformance Guide[3], includes the following text: "In addition to the variables specified for ADSL in the ADaMIG …, the sponsor should include multiple additional variables representing various important baseline subject characteristics / covariates presented in the study protocol." Therefore it seems that at least some baseline characteristics, the ones deemed "important," should be included in ADSL. But how do we determine which ones these are?

One option is to consider why each baseline characteristic is being derived. If it is to be used across most analyses, both safety and efficacy, then it probably makes sense to include it in ADSL. If only used in one analysis dataset, it might not be "important" enough to include in ADSL.

I once worked with a client that, in addition to a few baseline characteristics needed across all datasets, had more than 30 baseline characteristics that would be used just for one set of efficacy analyses. We elected to put those 30+ efficacy-only baseline characteristics in another (non-ADSL) ADaM dataset used just for efficacy. In fact, we created an intermediate dataset in a BDS structure, with traceability back to other ADaM datasets, just to hold this set of baseline values. A visual of that data flow is shown below:

Figure 3: Data Flow Option for Efficacy-Only Baseline Characteristics



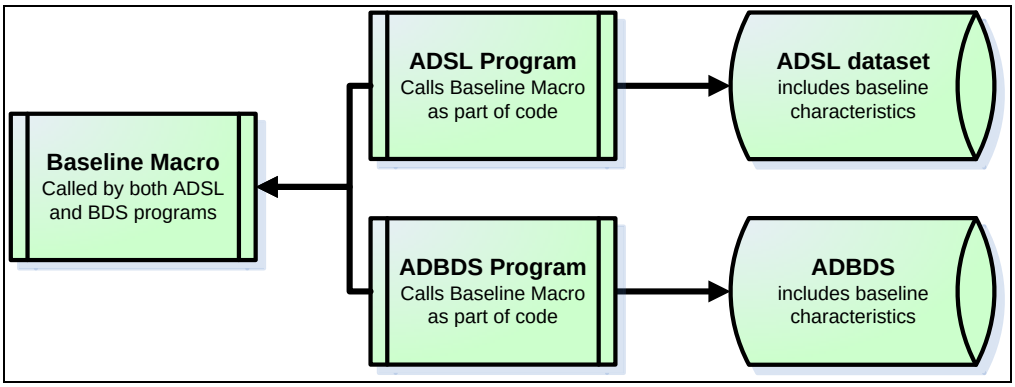
Whenever baseline characteristics are found outside of ADSL, such as described above, it would be helpful to communicate this to a reviewer, making use of the define and/or ADRG[2] documents.

It depends on whether the baseline characteristic needs to match the baseline derivation in a BDS dataset.

A simple linear data flow can be difficult when you need data from ADSL to derive BDS, but you also want BDS-derived data, such as a baseline value, as part of ADSL. This can lead to circular logic, where we a) create ADSL except for baseline characteristics, b) create BDS, including determining baseline, and then c) copy the BDS baseline results back to ADSL. Instead, when this type of a baseline characteristic is needed in ADSL, we want a solution that avoids circular references but also doesn't allow baselines in each dataset to get out of synch.

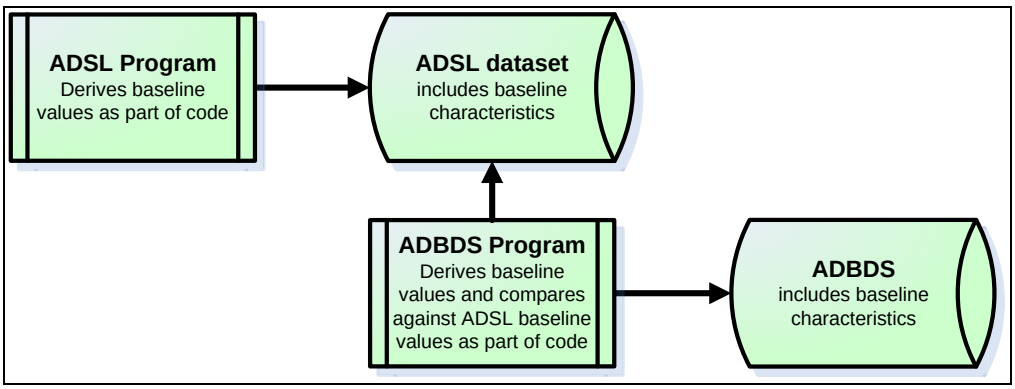
One solution is to use a macro to create baseline that is called twice, once when creating the baseline for ADSL and again when creating it for BDS. A visual of that flow is shown below:

Figure 4: Deriving Baseline via Macro code



Another solution is to keep the code separate, but include in the BDS code a check that the baseline values derived there match any baseline values that exist in ADSL. A visual of that flow is shown below:

Figure 5: Confirming Separately-Derived Baselines Match



Both of these two suggested methods, pulling out baseline derivation code into a macro that is called twice, or deriving baseline twice but including a check to confirm that they match, avoid the circular reference issue and follow a linear data flow.

It depends on whether the baseline characteristics are complicated or difficult to trace in ADSL.

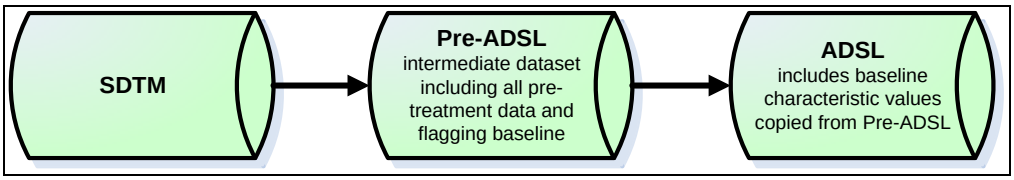
Unlike the more vertical ADaM structures, ADSL, with its one-record-per-subject structure, does not easily accommodate row-level data-point traceability variables such as SDTM --SEQ. For example, a baseline defined as the pre-dose value that represents the best case can be difficult to find when trying to trace back to a specific record in SDTM. When this type of a baseline characteristic is needed in ADSL, we want a solution that includes the fundamental principle of traceability.

One solution is to create, prior to ADSL, a vertical "BDS-like" intermediate dataset that includes this type of traceability. For the example stated above, where we want the best-case pre-dose value, we could:

1. Copy all pre-dose choices into an intermediate dataset, including the SDTM sequence number (using either the --SEQ or SRC* variables)
2. Flag in the intermediate dataset the row that will be used as baseline
3. Copy this baseline value from the intermediate dataset to ADSL

A visual of that flow is shown below:

Figure 6: Using an Intermediate Dataset to Derive Complex Baseline Values



This solution pushes the derivation of baseline to a predecessor intermediate dataset, a dataset that provides helpful traceability back to SDTM.

It depends on your analysis needs

As described above, there is no one right answer for all cases. The analysis need drives the decision on whether to include the baseline characteristics themselves in ADSL or in another ADaM dataset. When a baseline characteristic is needed in ADSL, we can avoid circular and non-traceable references by deriving the baseline outside of the ADSL program code via macro code or intermediate dataset code.

CATEGORY: STRUCTURE

3. DOES MY ADEX OR ADPK DATASET NEED TO USE THE BDS STRUCTURE?

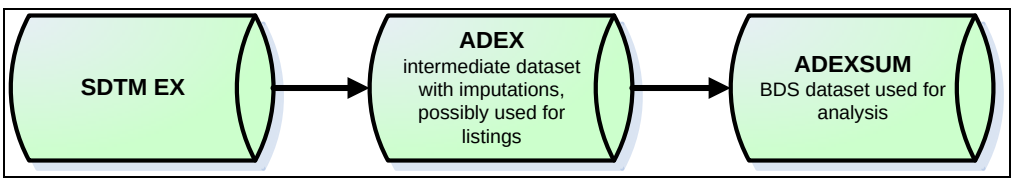
Answer: It depends on your analysis need.

The decision on whether to use the BDS structure for an analysis dataset depends on the analysis that will be done with the dataset, not about which SDTM domain the data is from. Some of our analysis needs, such as change from baseline and Time-to-Event, are easily mapped to a BDS structure. Others are not as clear.

It depends on whether ADEX will be used simply as an intermediate dataset

I've often seen a need for a dataset that brings in SDTM EX data and does some imputation in preparation for creation of an exposure summary dataset. This type of intermediate dataset itself isn't used for analysis, though often listings are generated from it. The exposure summary dataset is the one used for analysis, and can often be put into a BDS structure. A visual of that flow is shown below:

Figure 7: Intermediate exposure data with imputations



In this case, it can be most intuitive if the ADEX intermediate dataset is structured similar to the SDTM EX domain, adding some additional analysis variables such as imputed dates and dosages. Row-level traceability is inherent by simply carrying forward the SDTM EXSEQ variable. Using ADaM naming conventions for imputed dates and putting those numeric dates next to the character SDTM dates also provides traceability. In this case, ADEX is not in the BDS structure, but still follows the fundamental principles of ADaM.

It depends on whether following all the BDS rules for rows vs. columns would make the dataset unwieldy

Some of the rules regarding when to create rows in BDS would make an ADEX or ADPK too cumbersome to understand. For example, when you must use multiple rows to derive a value, this must be done using a new row rather than a column. This means a running cumulative dose at each time point can't technically be added as a column in BDS. Creating new rows to hold cumulative dose as a separate parameter however, might be difficult for a reviewer to understand.

As mentioned in the introduction, I firmly believe that the ADaM fundamental principles take precedence over any decision regarding ADaM structure. First and foremost, an ADaM dataset must follow the ADaM fundamental principles. When it makes sense for analysis needs, then we design the dataset to fit one of the ADaM structures. When following BDS would make a dataset too difficult to understand, we should consider other options.

As a rule of thumb, we must first determine whether following the rows vs. columns BDS rules makes a dataset too unwieldy to use. When it does so, we should consider deviating from that structure. In this case our dataset could still follow many of the BDS conventions, and we should make the effort to deviate from BDS only when needed. However, because it is not a true BDS, when documenting the dataset in the define file, it would be of class "ADAM OTHER".

It's worth noting here that at this time an ADaM sub-team is currently looking into PK analysis needs and considering whether another structure or modifications of a current structure would be useful here. ADaM continues to evolve to accommodate industry standard needs.

It depends on your analysis need.

As described above, there is no one right answer for all cases. The analysis need drives the decision on whether to create a BDS or other dataset structure. Before blindly trying to push all data into a BDS structure, first determine your analysis needs and then decide which structure makes the most sense to meet those needs. Conversely, don't assume you can't use a BDS structure without first trying to do so. In general, use the structure that best meets the fundamental principles of ADaM for the particular data and analysis needs.

Whenever a structure is needed that is not defined by ADaM, use as many ADaM conventions as possible and refer to the dataset with class "ADAM OTHER". For more information on ADAM OTHER, see John Troxell's PharmaSUG 2015 paper titled 'What is the "ADAM OTHER" Class of Datasets, and When Should it be Used?'[6].

4. CAN I USE THE ADAE STRUCTURE FOR CONCOMITANT MEDICATIONS OR MEDICAL HISTORY?

The ADaM Data Structure for Adverse Event Analysis[5] (ADAE) was written specifically for Adverse Events, but states in the introduction: "Adverse events are just one example of data that can use the structure described within this document. An ADaM sub-team is working to expand this to other data where there is no need for an analysis variable or parameter as would be seen in a BDS structure because records are simply counted for analysis. Example data for these types of analyses are concomitant medications and medical history." As of this writing, the Occurrence Data Structure is in final development to tackle these other types of analyses, and expected to soon be released for public use. Until then, there isn't a structure for concomitant medications or medical history data that is analyzed similarly to the standard adverse event analyses described in the ADAE document.

Answer: It depends on your analysis need.

First, we need to determine whether the concomitant medications or medical history data will be collected and analyzed in a similar way as the standard adverse event analyses described in the ADAE document:

- Is there a dictionary hierarchy of terms that are summarized by counting subjects within each level?
- Does the dictionary content from the SDTM data remain unchanged for analysis?
- Is there no need for AVAL or AVALC?

When all of the above questions are answered "Yes", then we want to model our data similar to ADAE. However, because ADAE is specific only for adverse events, any concomitant medications and medical history data in a similar structure would be of class "ADAM OTHER".

We don't always need to follow ADAE for concomitant medications and medical history data. For example, an analysis dataset isn't required when medical history data isn't analyzed. There may be other analysis needs, such as determining a change from baseline in concomitant medication dosages, which would likely use the BDS structure.

There is no one right answer for all cases. The analysis need drives the decision on which data structure to use, and even whether an analysis dataset is needed at all. Before blindly trying to push all concomitant medications and medical history data into an ADAE structure, first determine your analysis needs and then decide which structure makes the most sense to meet those needs. As mentioned previously, whenever a structure is needed that is not defined by ADaM, use as many ADaM conventions as possible and refer to the dataset class as "ADAM OTHER".

CATEGORY: METADATA

5. DO I NEED TO INCLUDE PARAMETER OR RESULTS METADATA?

Answer: It depends on your analysis need.

Most ADaM implementers are comfortable with analysis dataset and analysis variable metadata. Dataset and variable metadata have been used with submission data for many years, including with SDTM data.

Parameter metadata (formally "analysis parameter value-level metadata") and results metadata (formally "analysis results metadata") are not as familiar. Let's consider when it's appropriate or even necessary to use these additional types of metadata. Do you need a way to describe different derivation rules or controlled terminology that apply only to certain parameters? Do you have a way to describe how key analyses are created from your analysis datasets?

It depends on whether you have different variable metadata for different parameters.

In a BDS structure, different parameters can have different metadata needs. For example, if vital signs Height and Weight are captured in SDTM tests, it's often a 1:1 map from those tests and test codes to equivalent analysis parameters and parameter codes. However, within the same analysis dataset, we may also need to derive Body Mass Index (BMI) from height and weight. Here the derivation for parameter BMI is different than copied parameters height and weight. When creating metadata, we might initially consider something like this:

Variable	Label	Controlled Terms or Format	Derivation
AVAL	Analysis Value		For PARAMCD in ('HEIGHT' 'WEIGHT') set to VSSTRESN where TESTCD = PARAMCD For PARAMCD = 'BMI' derive as…
BASE	Baseline Value		AVAL where ABLFL = 'Y'
CHG	Change from Baseline		AVAL – BASE
VSSEQ	Sequence Number		For PARAMCD in ('HEIGHT' 'WEIGHT') set to VSSEQ where TESTCD = PARAMCD For PARAMCD = 'BMI' set to missing

Table 1. Parameter Metadata Within Derivation

Table 1 actually contains a form of parameter metadata, though it's hard to do much with it since, the parameter material is imbedded within a single large Derivation cell for each variable. An alternate method is to split out the parameter metadata into something like this:

Variable	Label	Controlled Terms or Format	PARAMCD	Derivation
BASE	Baseline Value			AVAL where ABLFL = 'Y'
CHG	Change from Baseline			AVAL – BASE
AVAL	Analysis Value		'HEIGHT', 'WEIGHT'	set to VSSTRESN where TESTCD = PARAMCD
AVAL	Analysis Value		'BMI'	Derive as …
VSSEQ	Sequence Number		'HEIGHT', 'WEIGHT'	set to VSSEQ where TESTCD = PARAMCD
VSSEQ	Sequence Number		'BMI'	<Missing>

Table 2. Parameter Metadata as a Separate Column

Table 2 contains exactly the same information as Table 1, just laid out a little differently. Specifying parameter metadata as shown in Table 2 allows us to more clearly show which metadata differs based on specific parameters.

Another benefit of splitting out parameter metadata, as shown in Table 2, is that it is sortable. Re-sorting by PARAMCD separates all the parameter-specific information from the non-parameter specific. This can aid in understanding the complete picture of each type of parameter, as shown below:

Variable	Label	Controlled Terms or Format	PARAMCD	Source/Derivation/Comment
BASE	Baseline Value			AVAL where ABLFL = 'Y'
CHG	Change from Baseline			AVAL – BASE
AVAL	Analysis Value		'BMI'	Derive as …
VSSEQ	Sequence Number		'BMI'	<Missing>
AVAL	Analysis Value		'HEIGHT', 'WEIGHT'	set to VSSTRESN where TESTCD = PARAMCD
VSSEQ	Sequence Number		'HEIGHT', 'WEIGHT'	set to VSSEQ where TESTCD = PARAMCD

Table 3. Re-Sorting Parameter Metadata by PARAMCD

Scenarios that could especially benefit from splitting out parameter metadata, such as in Table 2, include questionnaire data, where scores and sub-scores are derived; time-to-event data, with multiple complex parameters; and parameters that use different controlled terminology.

Parameter metadata exists, as part of both specifications and the define file, to help us communicate this type of information. Whenever variable metadata contains complex, lengthy, and complicated derivations, splitting out parameter metadata might help clarify it.

It depends on the reason for the analysis.

Results metadata is described in ADaM v2.1[1], though Version 1.0 of the Analysis Results Metadata Specification for Define-XML v2[7] contains more current content and examples.

The ADaM v2.1[1] states on page 21 "Analysis results metadata are not required. However, best practice is that they be provided to assist the reviewer by identifying the critical analyses, providing links between results, documentation, and datasets, and documenting the analyses performed."

Forward-thinking companies are starting to implement results metadata as part of their metadata repository, even if that is just well-controlled spreadsheets. This allows the metadata to not only be used for specifications, but also as input to the define file.

The Analysis Results Metadata Specification for Define-XML v2[7] shows the key components that results metadata convey:

Figure 8: Key Components of Results-Level Metadata

- Analysis Display metadata definitions
  - Analysis Result metadata definitions
    - Analysis parameter(s)
    - Analysis variable(s)
    - Analysis dataset(s)
      - Analysis variable(s)
      - Selection criteria
    - Documentation
    - Programming statements

Companies that don't have a metadata repository and automated tool to create the results metadata section of the define file can find it quite time consuming to manually produce results metadata. Since results metadata is currently not required by ADaM or FDA, those in this situation often choose to include this level of metadata for only the most important results, and sometimes not at all.

It depends on your analysis need.

As described above, there is no one right answer for all cases. Consider whether the information you need to convey differs across parameters, and if so, include parameter metadata to make that information clear. In order to determine how much results metadata to include, consider which analysis results are important to describe using results metadata, and whether you have (or can create and enforce) a metadata repository and tool that will help you create results metadata for your define file.

Parameter metadata is not difficult to write, and should be used whenever different parameters in the same dataset need different metadata. Results metadata is, at the time of this writing, not required and can be skipped entirely or included only for the most important analysis results.

6. DO I NEED TO INCLUDE AN ADRG?

Answer: It depends on your analysis need.

Neither ADaM 2.1[1] nor the ADaMIG 1.0[1] specifically state that an ADRG is required. The USFDA Technical Conformance Guide[3], includes the following text: "The preparation of an Analysis Data Reviewer's Guide … is recommended as an important part of a standards-compliant analysis data submission." It goes on to say "The ADRG provides FDA reviewers with context for analysis datasets and terminology, received as part of a regulatory product submission, additional to what is presented within the data definition file (i.e., define.xml). The ADRG also provides a summary of ADaM conformance findings." However, it doesn't go so far as to say that an ADRG is a required component of the submission.

We need to consider whether an ADRG would be a useful part of the submission package. Here are some of the sections the ADRG contains:

- Study Data Standards and Dictionary Inventory: This is where you describe which versions of standards are used, including CDISC, MedDRA, WHO Drug, etc.
- Analysis Considerations Related to Multiple Analysis Datasets: This is where you describe things like core variables used across all/most datasets, windowing, and imputation rules.
- Analysis Data Creation and Processing Issues: This is where you describe data dependencies, intermediate datasets, and data flow.
- Data Conformance Summary: This is where you describe how you checked your data for compliance, the issues that are flagged, and why you were not able to fix them.

This type of information is not easily found in the other components of the submission, namely the datasets and the define file. Therefore, the ADRG document provides a useful place to document material that is helpful for a reviewer. By using the official CSS ADRG[2], you're also providing the information in a familiar and consistent layout.

The Technical Conformance Guide[3] is focused on the "standards-compliant analysis data submission," and the ADRG was written specifically for ADaM data. For these reasons, some companies may choose to create an ADRG only for those studies which are CDISC-compliant. Considering all the helpful information that an ADRG contains, it seems like it'd be helpful to create one even for non-ADaM studies.

CONCLUSION

The answer "It depends on your analysis need" might initially seem disappointing, but in reality it's actually liberating. For example, rather than being tied to a specific set of structures that must be adhered to in all situations, we always consider what makes the most sense for our analysis needs. That means that if our analysis needs dictate it, we can put laboratory data into an occurrence data structure, or adverse events into a basic data structure. We can even use a structure that isn't BDS at all, yet still call it "ADaM" as long as it follows the ADaM fundamental principles and naming conventions.

When it feels like I'm trying to force a square peg into a round hole, I ask myself: "What would be the easiest for a reviewer to understand?" After all, making the data user-friendly is the intention of following the ADaM fundamental principles (traceable, clear and unambiguous, analysis-ready, usable by current tools, and accompanied by metadata).

Analysis data has always been, and will continue to be, more complex than tabulation data. With few exceptions, it's unlikely that analysis datasets will ever become fully automated. Instead, analysis dataset development will continue to require programmers, statisticians, and analysts who can understand the analysis need and make use of the industry standards to create a complete and understandable set of material to support the analysis results.

REFERENCES

[1] CDISC Analysis Data Model document (ADaM 2.1) and Analysis Data Model Implementation Guide (ADaMIG 1.0) are available for free download at www.cdisc.org.

[2] Analysis Data Reviewer's Guide (ADRG) template, instructions, and examples are available for free download at http://www.phusewiki.org/wiki/index.php?title=Optimizing_the_Use_of_Data_Standards.

[3] USFDA Technical Conformance Guide and Catalog are available for free download at http://www.fda.gov/ForIndustry/DataStandards/StudyDataStandards/default.htm.

[4] CDISC Study Data Tabulation Model Implementation Guide version 3.2 (SDTMIG 3.2) is available for free download at www.cdisc.org.

[5] CDISC Analysis Data Model (ADaM) Data Structure for Adverse Event Analysis is available for free download at www.cdisc.org.

[6] Troxell, John. 'What is the "ADAM OTHER" Class of Datasets, and When Should it be Used?' Proceedings of the PharmaSUG 2015 Conference.

[7] Analysis Results Metadata Specification for Define-XML v2 is available for free download at www.cdisc.org.

ACKNOWLEDGMENTS

Special thanks goes out to my fellow ADaM team members, especially John Troxell who helped review this paper. Any mistakes are mine.

CONTACT INFORMATION

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

Sandra Minjoe
Accenture Life Sciences
560 Mission St, Suite 1200
San Francisco, CA 94105
sandra.minjoe@accenture.com