PharmaSUG 2012 - Paper DS18

CDISC ADaM Application: Does All One-Record-per-Subject Data Belong in ADSL?

Sandra Minjoe, Octagon Research Solutions, Wayne, PA

ABSTRACT:

The CDISC (Clinical Data Interchange Standards Consortium) ADaM (Analysis Data Model) team has developed a one-record-per-subject structure called ADSL (Analysis Data Subject Level). The ADaM IG (Implementation Guide) version 1.0 describes many variables that are commonly used in ADSL, suggests that sponsors include additional variables that describe the subject's trial experience, but warns against including too much in this structure. This paper helps implementers determine what one-record-per-subject data should be included in ADSL, describes cases when it would be advisable to instead put this data in another structure, and gives examples/applications of other structures that could be used. Traceability is stressed, both across analysis datasets and back to SDTM data.

Some familiarity with SDTM and ADaM is assumed.

INTRODUCTION:

The ADaM IG describes two official data structures. The first is ADSL (Subject Level Analysis Dataset), a one-record-per-subject structure that contains subject-level attributes. Because of its structure, it can be merged onto any other clinical dataset, including other ADaM datasets and SDTM datasets.

The other structure described is BDS (Basic Data Structure), which contains one record per subject per analysis parameter, and as needed per analysis timepoint. The ADaM IG contains examples of data that fit well into BDS, including those for change-from-baseline and shift table analyses. The ADaM document titled "Examples in Commonly Used Statistical Analysis Methods" contains additional BDS examples.

Still in development are ADaM structures for occurrence data (like Adverse Events and Concomitant Medications) and multivariate data, neither of which fit into BDS. A draft ADAE document has been posted for review and at the time of this writing the final version is imminent, but it will not be discussed here because it isn't used for one-record-per-subject analysis. Multivariate data needs more than one analysis parameter on a row, could be one-record-per-subject, and will be addressed later in the document.

Some examples of one-record-per-subject data that are needed in a study include ADSL variables specified in the ADaMIG, baseline characteristics, cohort variables, efficacy variables, and protocol violations. At first glance it would seem that all of this one-record-per-subject data would thus belong in ADSL, simply due to its structure. However, some of our one-record-per-subject data really isn't a good fit for ADSL, as we'll see.

ADSL

Section 1.3 of the ADaM IG v1.0 makes a general statement about the content of ADSL: "It contains variables such as subject-level population flags, planned and actual treatment variables for each period, demographic information, stratification and subgrouping variables, important dates, etc. ADSL contains required variables (as specified in this document) plus other subject-level variables that are important in describing a subject's experience in the trial."

Section 3.1 of the ADaM IG lists these required subject-level variables for ADSL, including identifiers, demographics, population indicators, treatment variables, and trial dates. In addition, this section of the document warns against including some other types of one-record-per-subject variables in ADSL, such as key endpoints.

FDA (Food and Drug Administration) has also chimed in on the topic of ADSL content with their release of the CDER Common Data Standards Issues Document version 1.1/December 2011. Page 10 of that document states that they expect ADSL to also contain "multiple additional variables representing various important baseline patient characteristics".

While the definition of "important baseline characteristics", cohort variables, and disposition variables will obviously vary from study to study, we thus need to be sure to include all of this type of one-record-per-subject information in ADSL. So how do we know which of these variables to include in ADSL, other than all of those variables listed in ADaMIG section 3.1? Simple: we look at the tables we need to produce. ADSL should be able to produce the standard Demography, Baseline Characteristics, and Disposition tables found at the beginning of a study report. Thus any one-record-per-subject variables needed to produce these tables should be part of our ADSL data.

Traceability Issues In ADSL

As described in the ADaM version 2.1 document, traceability is not only one of the fundamental principles of ADaM, but also a cornerstone of ADaM. We must be able to trace back from the p-value on a table to the ADaM dataset and variable(s) used to derive it, to the SDTM dataset(s) and variable(s) used to create that ADaM dataset, and finally back to the annotated CRF, as shown:

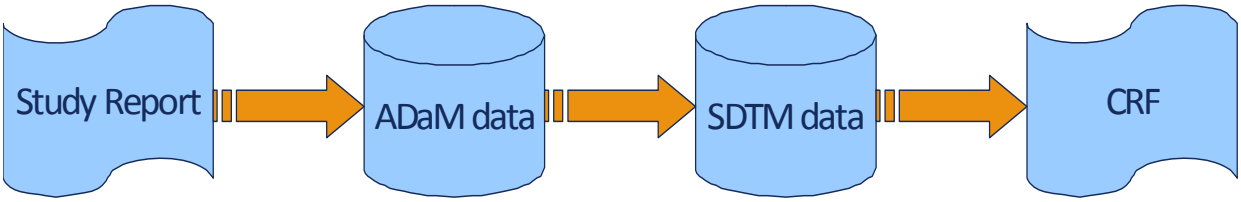


Figure 1: Overall ADaM Traceability

We strive for data point traceability, where we can point to specific record(s), not just the dataset and variable(s). With BDS data, data-point traceability is usually straightforward. For example, we can carry the --SEQ variable with us from SDTM and point to the exact SDTM row used to create the BDS row, as shown in Figure 2 below:

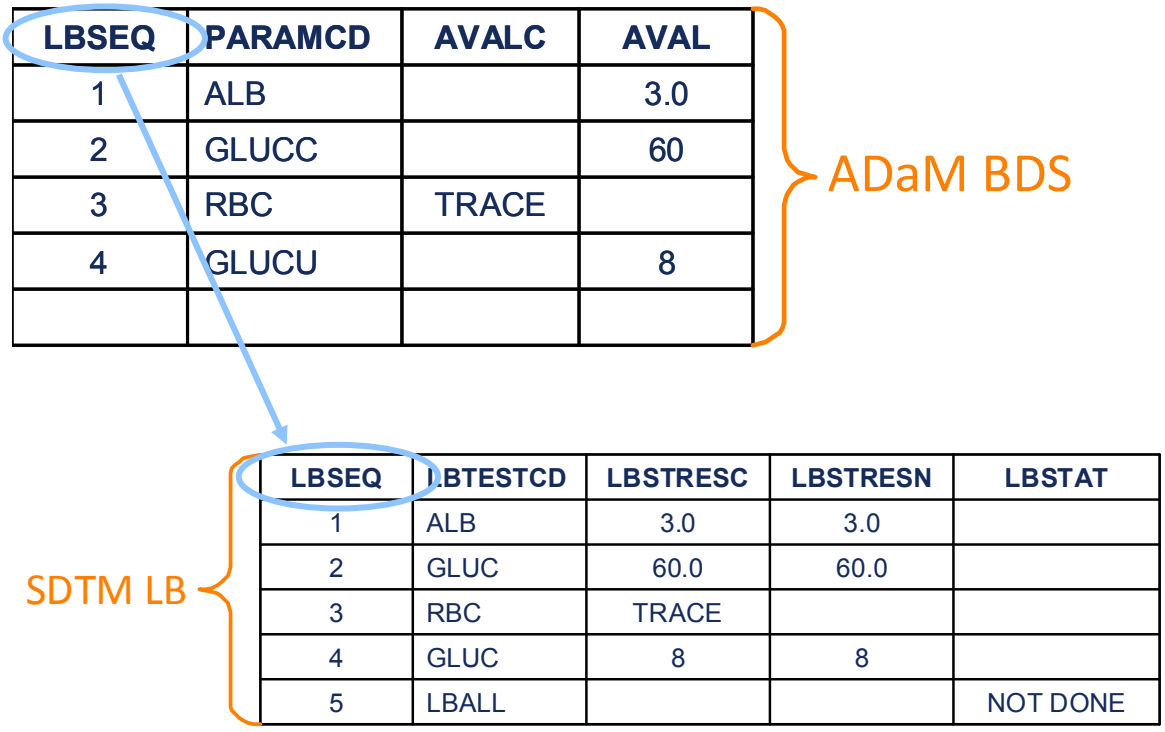
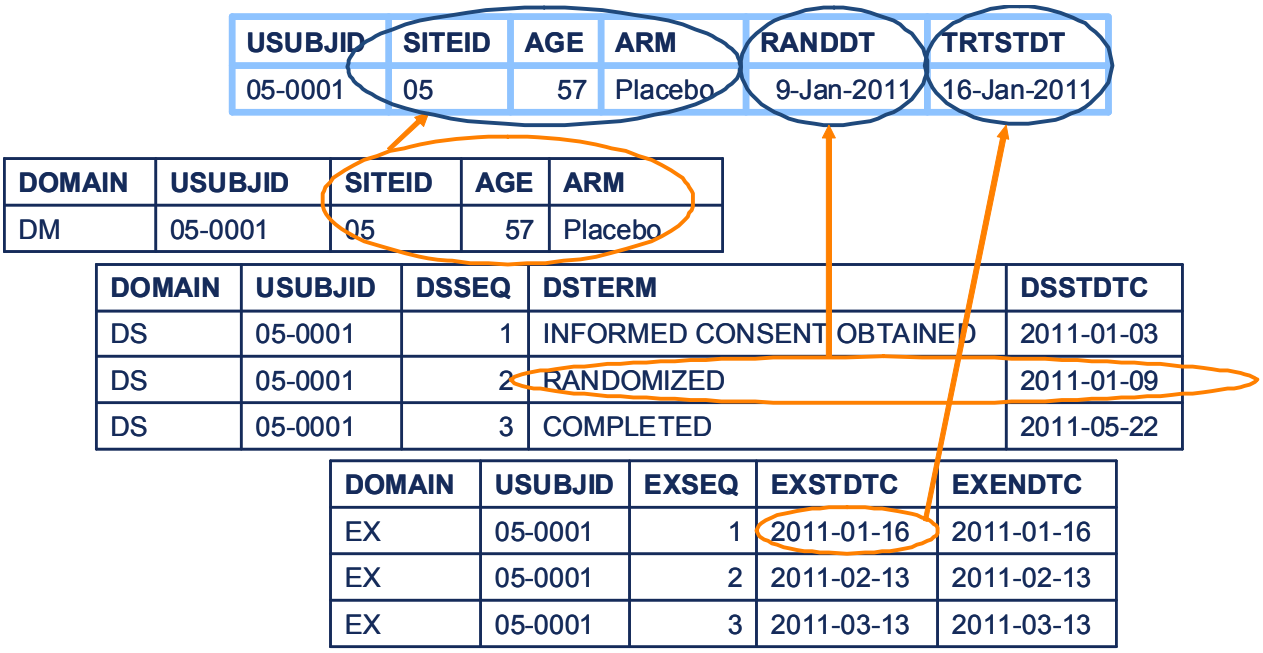


Figure 2: Example of BDS Data Point Traceability

One of the drawbacks of ADSL is that there isn't an easy way to provide this type of data point traceability. ADSL is a horizontal structure, so there is no place to put the sequence number from SDTM.

In many cases this lack of data-point traceability isn't an issue. Many variables included in ADSL are either from DM, which is structured as one-record-per-subject and doesn't include a sequence number. Other variables in ADSL have standard or straightforward derivations from BDS, so we can get by without this level of traceability.

In Figure 3, we see an example of pulling together data from DM, DS and EX to create some of our typical ADSL variables. It isn't too difficult here to determine where each variable in ADSL came from, even without the use of --SEQ or the SRC* variables:



USUBJID	SITEID	AGE	ARM	RANDDT	TRTSTDT
05-0001	05	57	Placebo	9-Jan-2011	16-Jan-2011

DOMAIN	USUBJID	SITEID	AGE	ARM
DM	05-0001	05	57	Placebo

DOMAIN	USUBJID	DSSEQ	DSTERM	DSSTDTC
DS	05-0001	1	INFORMED CONSENT OBTAINED	2011-01-03
DS	05-0001	2	RANDOMIZED	2011-01-09
DS	05-0001	3	COMPLETED	2011-05-22

DOMAIN	USUBJID	EXSEQ	EXSTDTC	EXENDTC
EX	05-0001	1	2011-01-16	2011-01-16
EX	05-0001	2	2011-02-13	2011-02-13
EX	05-0001	3	2011-03-13	2011-03-13

Figure 3: Example of ADSL data with no Traceability

Key features of this example:

- Data from the DM domain has the same variable names in ADSL, and because both are structured as one-record-per-subject there is no need for additional traceability information.
- There is (usually) only one randomization date per study, so a sequence number is probably not required even though we're copying from multiple-record-per-subject DS data.
- As we look at the records in EX, notice that we'd have to sort the treatment information to determine the first treatment start date and the copy that information without a sequence number to ADSL. Sorting and choosing the first record isn't a very complicated derivation, but it demonstrates how even with this simple example we're already starting to lose traceability.

Traceability with ADSL gets more complicated when we're talking about baseline variables. What is the definition of baseline? Is it the last observation before first treatment? If so, how do we handle partial dates that could be before or after that first treatment? What about a more complicated baseline derivation, such as the average of multiple values that happened prior to first treatment?

Even more complex, consider the definition of the per-protocol population flag, where often we must search across multiple SDTM datasets for a list of protocol violations to determine if the subject fails "per protocol".

As described earlier, both baseline characteristics and the per-protocol indicator flag belong in ADSL, but due to its one-record-per-subject structure there is no easy way to provide traceability back to SDTM. Because of this shortcoming, some have argued against including this data in ADSL. Instead let's consider options to add traceability to ADSL.

Traceability Solutions for ADSL

One solution to gain traceability for ADSL is to include derivation information as part of the metadata, such as in the define.pdf, reviewers guide, or even a program. This provides the necessary traceability and is helpful to anyone reviewing the data. However it can be complicated to read through long text fields in the define.pdf, link to the appropriate section of the reviewers guide, or weed through a program to find the parts related to the variable of interest.

Instead, consider the option of creating a dataset, prior to ADSL, to help with traceability. If we structure this dataset similar to BDS, so that it is one record per subject per analysis parameter, it can contain the traceability needed for ADSL. Note that because it is created directly from the SDTM data prior to creating ADSL, it won't have the required ADSL variables included, and thus won't be a true BDS. For these reasons, we'll refer to it as "BDS-like", meaning of the same structure as BDS but without all the required ADSL variables.

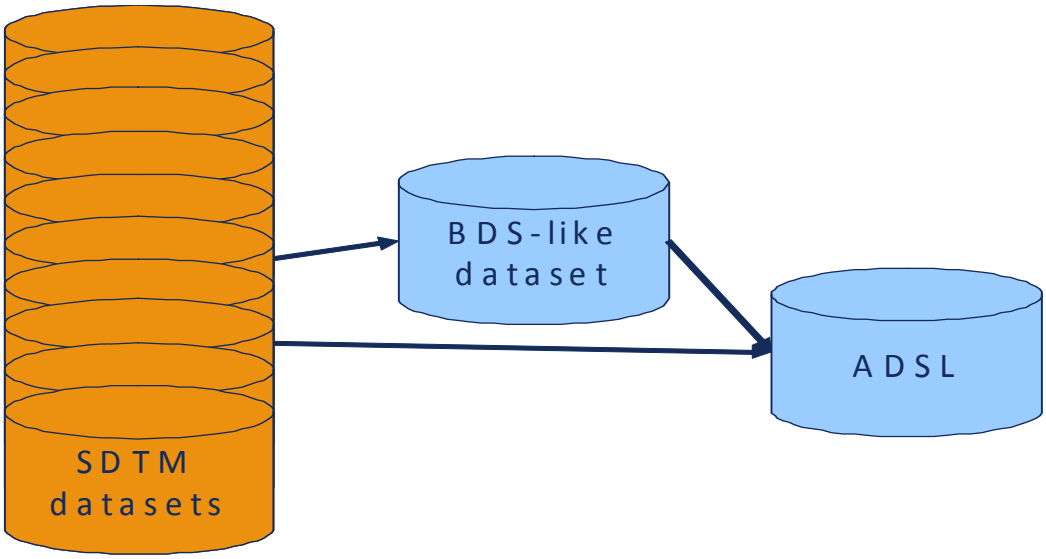


Figure 4: Traceability Solution Proposal

Example 1

As shown in Figure 3, the choice of exposure record used for TRTSDT in ADSL was not directly traceable back to EX. To provide that traceability using the proposed BDS-like method, we would first map the needed Exposure record to an interim dataset (we'll call it PADSL, as in "Pre-ADSL"), and then map from PADSL to ADSL.

Using the data from Figure 3, let's look at the first step, creating PADSL:

DOMAIN	USUBJID	EXSEQ	EXSTDTC	EXENDTC
EX	05-0001	1	2011-01-16	2011-01-16
EX	05-0001	2	2011-02-13	2011-02-13
EX	05-0001	3	2011-03-13	2011-03-13

EX

USUBJID	PARAMCD	AVAL	SRCDOM	SRCVAR	SRCSEQ
05-0001	TRTSTDT	16-Jan-2011	EX	EXSTDTC	1

PADSL

Figure 5: Demonstrating Traceability for ADSL Treatment Start Date (1 of 2)

We noted earlier that even though we have multiple records for the subject in EX, we need just the earliest EXSTDTC value, January 16, 2011, for our analysis variable TRTSTDT. That value (circled in green) is converted from character in SDTM EX to a numeric and stored in PADSL variable AVAL. The information circled in blue provides the traceability from EX to PADSL: the SDTM domain, name of the variable copied from, and sequence number. Dataset PADSL, in this BDS-like structure, thus supplies all the traceability we need to find the exact record copied from EX.

The next step is then creating ADSL from PADSL. How do we convey traceability when going from a more vertical structure to a horizontal one? It's actually quite simple here:

USUBJID	PARAMCD	AVAL	SRCDOM	SRCVAR	SRCSEQ
05-0001	TRTSTDT	16-Jan-2011	EX	EXSTDTC	1

PADSL

USUBJID	SITEID	ARM	AGE	RANDDT	TRTSTDT
05-0001	05	Placebo	57	9-Jan-2011	16-Jan-2011

ADSL

Figure 6: Demonstrating Traceability for ADSL Treatment Start Date (2 of 2)

Starting with the PASDL dataset from Figure 5, the same value of January 16, 2011 (circled in green) is copied to ADSL. Because TRTSTDT is both the PARAMCD in PADSL and the name of the variable in ADSL (circled in fuchsia), this provides the traceability between those two datasets.

Expanding on this idea, TRTENDT could also be mapped in a similar way: first copy the latest EXENDTC value to PADSL variable AVAL for the PARAMCD = TRTENDT, and copy to the PADSL SRC* variables all the information we need for traceability. Then copy from PADSL the content of AVAL for the PARAMCD = TRTENDT to the variable TRTENDT in ADSL. Now both of our derived ADSL variables, TRTSTDT and TRTENDT, have a corresponding PARAMCD in PADSL, and PADSL includes SRC* variables to provide the traceability back to the SDTM data. As PADSL gets longer with more PARAMCDs, ADSL gets wider with more variables.

This is a pretty simplistic example, included here only to show the process. We're not trying to suggest that we always need to handle the derivation of TRTSTDT and TRTENDT this way.

Example 2

Consider, however, the more complicated situation of generating the per-protocol population flag PPROTFL, where we must comb through many SDTM datasets looking for various protocol violations. To illustrate, we'll use just three different protocol violations: whether the subject completed the study as planned, whether they took all doses of study drug, and whether they failed any of the inclusion or exclusion criteria. This then involves reviewing SDTM domains DS, EX, and IE.

For one subject, we can determine that PPROTFL should be Y based on the following SDTM data:

Completed as planned

DOMAIN	USUBJID	DSSEQ	DSTERM	EPOCH	DSSTDTC
DS	05-0001	2	RANDOMIZED		2011-01-09
DS	05-0001	3	COMPLETED	TREATMENT PHASE	2011-05-22

DOMAIN	USUBJID	EXSEQ	VISIT	EXSTDTC	EXENDTC
DS	05-0001	1	Month 1	2011-01-16	2011-01-16
DS	05-0001	2	Month 2	2011-02-13	2011-02-13
DS	05-0001	3	Month 3	2011-03-13	2011-03-13

No IE records for the subject

Took all 3 study drug doses

Figure 7: Traceability Needs for ADSL Per-Protocol Population Flag

When creating our pre-ADSL dataset, we can take advantage of the BDS features for bringing in data from multiple domains and deriving new parameters as follows:

DOMAIN	USUBJID	DSSEQ	DSTERM	EPOCH	DSSTDTC
DS	05-0001	1	INFORMED CONSENT OBTAINED		2011-01-03
DS	05-0001	2	RANDOMIZED		2011-01-09
DS	05-0001	3	COMPLETED	TREATMENT PHASE	2011-05-22

DS

DOMAIN	USUBJID	EXSEQ	VISIT	EXSTDTC	EXENDTC
EX	05-0001	1	Month 1	2011-01-16	2011-01-16
EX	05-0001	2	Month 2	2011-02-13	2011-02-13
EX	05-0001	3	Month 3	2011-03-13	2011-03-13

EX

USUBJID	PARAMCD	AVALC	SRCDOM	SRCVAR	SRCSEQ	PARAMTYPE
05-0001	TRTDISC	2011-05-22	DS	DSSTDTC	3	
05-0001	TRTMO1	2011-01-16	EX	EXSTDTC	1	
05-0001	TRTMO2	2011-02-13	EX	EXSTDTC	2	
05-0001	TRTMO3	2011-03-13	EX	EXSTDTC	3	
05-0001	IECRIT	None	IE			Derived
05-0001	PPROTFL	Y				Derived

PADSL

Figure 8: Demonstrating Traceability for Per-Protocol Population Flag (SDTM and PADSL)

In the above example, we would first do the following data collection in PADSL:

- Copy from DS information about treatment phase discontinuation to determine if the subject completed the treatment phase as planned. (Circled in blue.)
- Copy from EX information about the dosing to determine if the subject received all doses as planned. (Circled in fuchsia.)
- Copy from IE information about inclusion or exclusion criteria to determine if the subject failed any criteria. (No records for this subject.)

In practice, we may not actually have data from DS, EX or IE to copy, such as seen here with IE. In those cases we'd have to make some assumptions, such as:

- If no treatment phase discontinuation record exists in DS, this implies that the subject did not complete the treatment phase as planned. Derive a record with PARAMCD=TRTDISC and AVAL=missing. Because this is derived, set PARAMTYP to DERIVED.
- If no dosing records exist in EX, this implies that the subject did not receive any treatment.
- If no records exist in IE, this implies that the subject did not fail any inclusion or exclusion criteria. Derive a record with PARAMCD=IECRIT and AVAL=missing. Because this is derived, set PARAMTYP to DERIVED. (Boxed in green.)

After all this information is compiled and any derivations made for missing data, we can then derive in PADSL the parameter PPROTFL, again using BDS rules. The following logic will handle that derivation:

- If the subject has non-missing AVAL for PARAMCD=TRTDISC, TRTMO1, TRTMO2, and TRTMO3, plus a missing AVAL for PARAMCD=IECRIT, then set AVAL of PPROTFL to Y; otherwise, set to N. Because this is a derived parameter, we must set PARAMTYP to DERIVED.

As in Example 1, traceability from PADSL to ADSL is straightforward, because the parameter and analysis value from PADSL become the variable name and content in ADSL:

USUBJID	PARAMCD	AVALC	SRCDOM	SRCVAR	SRCSEQ	PARAMTYPE
05-0001	TRTDIS	2011-05-22	D	DSSTDT	3	
05-0001	TRTMO1	2011-01-16	E	EXSTDT	1	
05-0001	TRTMO2	2011-02-13	E	EXSTDT	2	
05-0001	TRTMO3	2011-03-13	E	EXSTDT	3	
05-0001	IECRIT		IE			DERIVE
05-0001	PPROTFL	Y				DERIVE

PADSL

USUBJID	PPROTFL
05-0001	Y

ADSL

Figure 9: Demonstrating Traceability for Per-Protocol Population Flag (PADSL and ADSL)

With this technique, PASDL can contain many different parameters: some that translate directly to ADSL variables and some are used only to derive other parameters. Both types of parameters add value in terms of traceability.

Using this method of a pre-ADSL BDS-like structure means that much of the information needed to describe derivations can be determined strictly by viewing datasets and their associated metadata, rather than relying on external text and programs. Reviewers are already looking at datasets and metadata, so this could make the review much more straightforward.

NON-ADSL OPTIONS:

The solution proposed above handles the situation of highly derived variables that belong in ADSL. But what about other one-record-per-subject variables like primary efficacy that, per ADaMIG, don't belong in ADSL? Depending on our analysis needs, there are a couple different solutions: using the BDS structure, or creating another one-record-per-subject analysis dataset. Let's look at when each is appropriate.

BDS for One-Record-Per-Subject Data

When we're used to working with data in a more horizontal structure, BDS might seem difficult to use for analysis. It turns out, however, that data structured with analysis variables as rows works pretty much the same way in analysis as when structured with analysis parameters as columns. For most of our analysis needs, we have exactly one analysis variable/parameter per analysis. Sure, we may need to include cohort or censoring variables, but these are not the actual analysis variable/parameter. Regardless of the structure in which it is stored, when we're preparing a dataset for analysis we then typically use only one of our analysis variables/parameters. Whether we're sub-setting rows within a data set to get one analysis parameter, or sub-setting columns to get one analysis variable, the end result that is pushed into the statistical procedure looks virtually the same.

For example, consider the following two full analysis structures on the left in Figures 10 and 11, each containing the same information, and how they are subset to use for analysis. The structure shown in Figure 10 is horizontal with three analysis variables. The structure shown in Figure 11 is vertical with three analysis parameters. Each has two analysis cohort variables. The data on the right for each figure shows what it would look like after sub-setting to just the first variable/parameter and the first cohort variable, all that we'll need for one of our statistical procedures.

Horizontal Dataset with 3 analysis variables

USUBJID	AVAL1	AVAL2	AVAL3	COHORT1	COHORT2
1	25	82.9	-7	Y	3
2	30	77.2	0	Y	5

Data used in one procedure:

USUBJID	AVAL1	COHORT1
1	25	Y
2	30	Y

Figure 10: Sub-setting Horizontal data to just the data used in a procedure

Vertical Dataset with 3 analysis parameters

USUBJID	PARAM	AVAL	COHORT1	COHORT2
1	AVAL1	25	Y	3
1	AVAL2	82.9	Y	3
1	AVAL3	-7	Y	3
2	AVAL1	30	Y	5
2	AVAL2	77.2	Y	5
2	AVAL3	0	Y	5

Data used in one procedure:

USUBJID	AVAL	COHORT1
1	25	Y
2	30	Y

Figure 11: Sub-setting Vertical data to just the data used in a procedure

As we can see, the data used in the analysis procedure are virtually the same, whether initially from a horizontal or vertical structure.

It would seem, then, that it should make no difference whether we start with a vertical or horizontal analysis structure. There is one important advantage to the vertical structure, as we saw earlier for the BDS-like pre-ADSL structure, and that is traceability. This is actually why ADaM uses the more vertical structure for BDS: it has all the advantages of a horizontal structure, plus it allows for data-point traceability.

Thus the solution for one-record-per-subject analysis data that doesn't belong in ADSL is usually the BDS structure.

Multivariate analysis needs

There is an exception to the generalization that we can use the BDS structure for our one-record-per-subject non-ADSL analysis: multivariate analysis requires more than one analysis variable for the procedure. Because the BDS structure has only one analysis parameter per row, it won't work for multivariate analysis.

Although we can't use BDS directly to produce multivariate analysis results, we don't simply scrap the BDS concept completely. The ADaM document "Examples in Commonly Used Statistical Analysis Methods" describes how BDS can still be useful. Deriving data for multivariate analysis just becomes a two-step process:

1. Collect and derive analysis parameters in the BDS structure as if each parameter were to be used individually for analysis. (Note that in some cases, analysis like this might need to be done.)
2. Transpose the BDS parameters needed for multivariate analysis to a second more horizontal structure to make it multivariate analysis-ready.

This two-step process takes advantage of the traceability features of BDS, yet still allows our analysis dataset to be one statistical procedure away from our analysis results. The process is similar to what was described earlier for generating ADSL data from a BDS-like pre-cursor, except now we can use a true BDS structure as a pre-cursor.

A simple example is shown here:

USUBJID	PARAM	AVAL	COHORT1	COHORT2
1	AVAL1	25	Y	3
1	AVAL2	82.9	Y	3
1	AVAL3	-7	Y	3
2	AVAL1	30	Y	5
2	AVAL2	77.2	Y	5
2	AVAL3	0	Y	5

BDS structure

Transpose so that each analysis parameter becomes a column

USUBJID	AVAL1	AVAL2	AVAL3	COHORT1	COHORT2
1	25	82.9	-7	Y	3
2	30	77.2	0	Y	5

Non-BDS structure

Figure 12: Transposing BDS data in order to perform Multivariate Analysis

For a more elaborate example and further explanation on how to create a multivariate analysis dataset, refer to the ADaM document "Examples in Commonly Used Statistical Analysis Methods".

Also note that there is an ADaM sub-team currently working on a more formal structure for multivariate analysis data. This group is tentatively calling that structure MVDS (Multi-Variate Data Structure).

CONCLUSION:

The answer to our original question, "Does All One-Record-per-Subject Data Belong in ADSL?" is, in fact, no. We've described in this document several different ways to handle one-record-per-subject analysis data:

- We can include it in ADSL, as long as it fits the intent of that dataset. By this we mean that it is one of the types of data included in Section 3.1 of the ADaM IG, or is other subject level data, such as baseline characteristics, needed to create tables such as Demographics, Baseline Characteristics, and Disposition.
- When one-record-per-subject data needed for ADSL is highly derived, also create a pre-ADSL dataset in a BDS-like structure. This pre-ADSL dataset wouldn't be used directly to produce analysis results, but it provides traceability back to SDTM.
- Most other (non-ADSL) one-record-per-subject data fits nicely into a BDS structure. Including data in this vertical structure provides traceability and is easily subset for most of our analysis needs.
- For multivariate analysis, which needs multiple analysis variables on a single record, we should first create a BDS structure to hold the multivariate parameters and any predecessor parameters needed for traceability, and then transpose the parameters needed for multivariate analysis so that all the necessary information is available as columns.

We can take advantage of the BDS (or a BDS-like) structure, with its simple traceability features, to help us with our one-record-per-subject analysis data needs. The following figure shows a flow diagram of how we can generate all of our one-record-per-subject analysis-ready data, using BDS-like, ADSL, BDS, and MVDS structures. It demonstrates that we can create many different analysis data structures, each fitting on our one-record-per-subject analysis needs, all while still maintaining traceability back to SDTM:

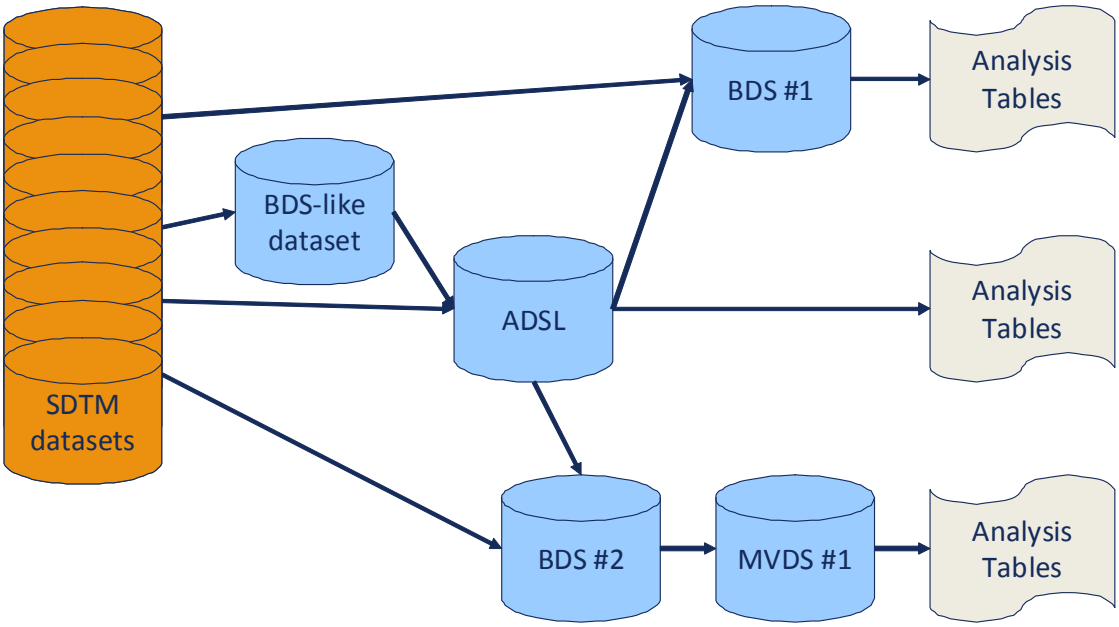


Figure 13: Data Flow to Create One-Record-Per-Subject Data that Enables Traceability

REFERENCES:

"Analysis Data Model (ADaM)". http://www.cdisc.org/adam. The current version is downloadable from web page and available to CDISC members and non-members (access date 12Jan2011).

"Analysis Data Model (ADaM) Examples in Commonly Used Statistical Analysis Methods". http://www.cdisc.org/adam. The current version is downloadable from web page and available to CDISC members and non-members (access date 13Feb2012).

"Analysis Data Model (ADaM) Implementation Guide". http://www.cdisc.org/adam. The current version is downloadable from web page and available to CDISC members and non-members (access date 12Jan2011).

"CDER Common Data Standards Issues Document". http://wcms.fda.gov/FDAgov/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM248635. The current version is downloadable from this FDA web page. A standard web search on the document title will quickly find the document (access date 21Dec2011).

ACKNOWLEDGEMENTS:

The author would like to acknowledge Nate Freimark. He presented a paper titled "Patient Evaluability in the CDISC World" at the 2009 CDISC Interchange that described the process of creating a pre-ADSL dataset in a BDS-like structure. This presentation is not included in the References section above only because it is not publically available.

CONTACT INFORMATION:

Your comments and questions are valued and appreciated. The author can be reached at:

Sandra Minjoe
Octagon Research Solutions
585 East Swedesford Rd
Wayne, PA 19087 U.S.A.
sminjoe@octagonresearch.com