Avoiding Sinkholes: Common Mistakes During ADaM Data Set Implementation

Richann Watson, Experis, Batavia, OH

Karl Miller, inVentiv Health, Lincoln, NE

ABSTRACT

The ADaM Implementation Guide was created in order to help maintain a consistency for the development of analysis data sets in the pharmaceutical industry. However, since its inception we have seen issues with guideline non-conformance which can impede this development process and carry impacts that are felt down-stream in subsequent processes. When working with ADaM data sets, non-compliance and other related issues are likely the number one source for numerous hours of re-work; not only creating unnecessary additional work for the data sets themselves, but also for reports, compliance checks, the Analysis Data Reviewers Guide (ADRG), etc. all the way down to the ISS/ISE processes. Considering this breadth of impact, one can see how devastating these sinkholes can be. Like any sinkhole, there is a way out of it but it is a long, tedious process that will consume a lot of resources and it is always better to avoid the sinkhole entirely. This paper will assist you in creating compliant ADaM data sets, provide the reasoning on why you should avoid these sinkholes, all of which will help minimize re-work and likely eliminate the need for additional work.

INTRODUCTION

With the FDA now requiring data to be submitted using CDISC standards, companies are now working to get up to speed and make all their analysis data sets ADaM compliant. However, the ADaM Implementation Guide (ADaM IG) can be confusing and may lead to misinterpretation which can cause a non-compliant data set. Without a full understanding of the ADaM IG, creation of ADaM data sets can go awry and you can end up with data sets that can cause problems downstream.

COMMON MISTAKES

Below we will walk through some of the more common mistakes we have seen. In addition we will provide some recommendations that will make the data set CDISC compliant. The order in which issues are listed does not by any means indicate severity of the issue. All issues should be addressed so that the data sets are CDISC compliant.

NOT FOR LISTINGS

One of the most common mistakes we have encountered is the creation of an ADaM data set for the generation of a listing. Listings are considered to be a 'dump' of the data and not analysis, and since a listing is not considered analyses, then there is no need to create an ADaM data set. There is no requirement that there be a one to one relationship from SDTM to ADaM. ADaM data sets are determined based on analysis needs so it is possible for multiple SDTM domains to feed into one ADaM data set, or for an ADaM data set to be created from other ADaM data sets.

Recommendations

- The ideal approach for creation of listings is to use the corresponding SDTM domain. Below are some situations that can help determine if it is ideal to use the SDTM domain
    - If the listing is coming from one data source and there are no derived variables, then the SDTM domain should be the source of the listing.
    - If the listing is coming from one data source and only needs to include the population flags from ADSL, then the merge of ADSL to the SDTM domain can take place in listing program.
- If the listing is coming from one data source but study day needs to be re-calculated based on a treatment or analysis period date found in ADSL, then consider creating an ADaM data set that does the merge of ADSL to the SDTM domain and the re-calculation, since FDA reviewers are not programmers and may not be comfortable merging data sets and deriving variables.
- In cases where listings need to be produced to support a table that was based off of derived parameters, the ADaM data set used to create the table should also be used to produce the listing. For example, when assessing time to event, the ADTTE data set could be used to generate the listing rather than trying to pull all the various data sources into the listing program.

NOT SDTM +

Some approaches we have seen take the existing SDTM domain and either appends the supplemental qualifier domain to the parent domain and refers to that as 'OTHER' or they rename the SDTM variables to corresponding ADaM variables. For example, the following variables would be renamed accordingly: --TESTCD to PARAMCD, --TEST to PARAM and --STRESN(C) to AVAL(C).

The Basic Data Structure (BDS) has required variables (i.e., PARAMCD, PARAM and AVAL/AVALC) and the incorporation of these variables into a data set without considering the other rules does make the data set a BDS. There are specific rules that govern the creation of a BDS data set. In addition, there are principles that need to be adhered to when creating ADaM data sets.

Recommendations

- Determine what the analysis needs are in order to figure out what the correct data structure should be. Just because the SDTM domain is not a Findings class does not mean the analysis data would be classified as 'OTHER'. The majority of the analyses performed can be handled using the BDS. There are specific rules that govern the creation of a BDS data set, which are specified in the ADaM IG.
- If the analysis involves the counting of incidences/events, then the Occurrence Data Structure (OCCDS) should be used and the pre-defined variables should be implemented where applicable.
- If the analysis does not warrant the use of either BDS or OCCDS, then a data set that fully supports the generation of the analysis and is of the structure 'OTHER' can be created as long as the data set adheres to the four fundamental principles:
    - Clear and unambiguous communication (i.e., traceability; data point and/or metadata traceability)
    - Contains metadata
    - Analysis ready (i.e., produce the desired analysis in one procedure with a subsetting where clause)
    - Machine readable by commonly used software

KNOW WHAT IS BEING ANALYZED

Often the development of ADaM data set specifications begins without having a complete picture of what is being analyzed. Without knowing what is being analyzed, it is difficult to make the data sets analysis ready, which is one of the fundamental principles. The lack of knowledge in regards to the analysis can lead to hours of re-work and lots of frustration. In addition, it can lead to a violation of analysis ready principle because most programmers may initially find it easier to just make the updates in the table /figure (TF) programs rather than go back and update the ADaM data set specifications and the ADaM data set programs. However, this is less than ideal. Not only is a fundamental principle being violated, if there are multiple TF outputs that require the same logic and the logic is implemented in the TF programs then each program would need to be updated every time the logic changes. Furthermore, it would be easy to overlook a criterion in one TF program, or maybe approach it in a different way that would cause the results between the two outputs to be out of alignment. Thus, incorporating the logic directly into the TF programs would require a lot of cross-checking to make sure the programs and results are in sync and in the end this would cause more work than if it was just incorporated into the ADaM data set from the start.

Recommendations

- Ask the statistician for the TF shells.
- If TF shells are available, go through each one to see what data and what type of analyses are needed, and create a draft annotation of the shells that can be used as a guideline for defining the necessary ADaM data sets.

VARIABLE HARMONIZATION

A common issue that is typically encountered is the recalculation of AGE and other variables that are found in an SDTM domain. ADaM adheres to the "'same name, same meaning, same values' principle of harmonization"[1] In other words, if the variable exists in an SDTM domain then it should be copied without modifications.

Recommendations

- If a variable in SDTM requires a recalculation, then the original variable should be copied as is and a new variable with the re-calculation created to capture the new value.

Illustration

Table 1 and Table 2 illustrate the concept of variable harmonization. In the DM domain in Table 1, AGE is based off of the informed consent. However, the SAP indicates that for analyses the age needs to be based off of the first treatment date. Even if the recalculation of age may not change, since the definitions for age are different, both the original SDTM AGE variable and the new ADaM AAGE variable should be retained as illustrated in Table 2.

DM				
USUBJID	**RFICDTC**	**BRTHDTC**	**AGE**	**AGEU**
ABC-001-001	2016-12-02	1972-07-24	44	YEARS
ABC-001-001	2016-11-16	1976-11-24	39	YEARS

EX	
USUBJID	**EXSTDTC**
ABC-001-001	2017-01-05
ABC-001-001	2016-12-20

Table 1 DM and EX for Variable Harmonization Illustration

ADSL				
USUBJID	**BRTHDTC**	**TRTSTDT**	**AGE**	**AAGE**
ABC-001-001	1972-07-24	05JAN2017	44	44
ABC-001-001	1976-11-24	20DEC2016	**39**	**40**

Table 2 ADSL Variable Harmonization Illustration

RE-CREATING PRE-DEFINED VARIABLES

As the ADaM IG goes through updates, new variables are created that become part of the standards. If a concept has been pre-defined in the ADaM IG, then the associated variable must be used rather than create a user-defined variable. Per section 3.1.1 item 4 in the ADaM IG

> "When an ADaM standard variable name has been defined for a specific concept, the ADaM standard variable name must be used, even if the content of an ADaM variable is a direct copy of an SDTM variable. For example, in the creation of ADLB, even if AVAL is just a copy of LBSTRESN then dataset must contain AVAL."[2]

Furthermore when creating user-defined variables, the ADaM IG has defined variable naming fragments that must be used for specific concepts and only for those concepts. Per section 3.1.5 of the ADaM IG

> "… a list of standard suffix fragments (i.e., variable name fragments used as the last part of a variable name) that are required when naming variables in ADaM datasets …For these fragments, it is a requirement that the appropriate fragment be used whenever the concept applies and the fragment is reserved to be used only for that corresponding concept." [2]

There are additional naming fragments that can help with the creation of user-defined variables. The appropriate fragment that best conveys the concept of the variable within the variable naming conventions should be used when naming a variable.

Recommendations

- Become familiar with pre-defined variables
- Become familiar with pre-defined naming fragments defined in Section 3.1.5 of the ADaM IG.

V5 TRANSPORT FILES

There are certain regulations that need to be followed when creating ADaM data sets. It is a requirement that the data sets adhere to SAS® Version 5 transport rules.

- Variable name
  - No longer than 8 characters
  - Must start with a letter
  - Must only contain letters, numbers and underscore
- Variable label
  - No longer than 40 characters
- Value
  - No longer than 200 characters.

In addition, to the variables having to adhere to SAS V5, the value of PARAMCD should also be no longer than 8 characters.

Recommendations

- Review the data set to confirm that the variables and values of PARAMCD follow the rules required for a transport file.

MAINTAINING TRACEABILITY

Traceability is one of the fundamental principles of CDISC. It instills confidence in the results. Without traceability there is no way to link back to the source data and no way to ensure that what was said to be done was actually done. By maintaining traceability, you are making the data transparent by showing the relationship between the analysis results and the ADaM data sets and the SDTM domains.

Recommendations

- Submit all analysis data sets even if they are intermediate data sets and will not be used to produce the actual analyses. Intermediate data sets are great to help gather all information in one place especially when complex computations are involved. If an intermediate data set is needed, then it has to be submitted.
- Data point traceability should be used when possible so it is readily evident what the predecessor record(s) is.
- Metadata traceability should be included so that the user / reviewer understands the relationship of the analysis data to the source data.

Illustration

There various ways to achieve data point traceability. One way is to include --SEQ or ASEQ if the source data set has a sequence variable (Table 3).

USUBJID	LBSEQ	VISITNUM	PARAMCD	AVAL
ABC-001-001	***215***	1	AST	25
ABC-001-001	***216***	1	ALT	40
ABC-001-001	***217***	1	GGT	21
ABC-001-001	***218***	1	ALP	65

Table 3 --SEQ from SDTM Domain to Illustrate Data Point Traceability

If the data comes from multiple sources, data point traceability may still be achieved with the incorporation of the SRC variables. The SRCDOM will indicate the SDTM domain or ADaM data set that the record originated in while SRCVAR indicates the variable that is used to populate the new data set and SRCSEQ is used to indicate the record that is used. Note that if the variable is the same for all records then there is no need to include SRCVAR. Also, if the source data set is one record per subject there may not be a –SEQ variable thus SRCSEQ would be left null. In Table 4 the daily dose can either come from the Drug Accountability domain (DA) or the Exposure domain (EX).

USUBJID	PARAMCD	ADT	AVAL	SRCDOM	SRCVAR	SRCSEQ
ABC-001-001	DLYDOSE	01JAN2014	20	***DA***	***DASTRESC***	***14***
ABC-001-001	DLYDOSE	02JAN2014	40	***DA***	***DASTRESC***	***15***
ABC-001-001	DLYDOSE	01FEB2014	20	***EX***	***EXDOSE***	***161***
ABC-001-001	DLYDOSE	02FEB2014	20	***EX***	***EXDOSE***	***162***

Table 4 SRC Variables to Illustrate Data Point Traceability

The inclusion of key SDTM variables can also lend to data point traceability. In some scenarios it is ideal to include certain SDTM variables to easily link the record in the ADaM data set back to the source data. For example, in Table 5 the data contains two records for LBSEQ = 5 and LBSEQ = 9 with different values for AVAL. With just LBSEQ used to link back to SDTM, it is not readily evident which record was the original record. The incorporation of LBSTRESC allows for a comparison with AVAL to determine which record differs from the source data. In addition, the data set contains BQL values (e.g., --STRESC = '<xx'). In the ADaM data set, a numeric value is expected and BQL needs to be converted but the original value must also be retained. Since AVAL and AVALC must maintain a 1-1 match, then AVALC cannot be used to keep the original value, thus the corresponding variable from SDTM should be carried over to the ADaM data set.

USUBJID	LBSEQ	VISITNUM	PARAMCD	AVAL	LBSTRESC	DTYPE
ABC-001-001	5	1	DBILISI	3.1	3.1	
ABC-001-001	5	1	DBILICN	**0.1813**	**3.1**	
ABC-001-002	9	1	DBILISI	**1**	**< 2**	**BQL**

USUBJID	LBSEQ	VISITNUM	PARAMCD	AVAL	LBSTRESC	DTYPE
ABC-001-002	9	1	DBILICN	**0.0585**	**< 0.117**	**BQL**

Table 5 SDTM Variable to Illustrate Data Point Traceability

Since the last three rows in the data in Table 5 have some kind of formula that is used to populate AVAL, it is necessary to have metadata to show the relationship of the record to the source data. How the metadata is captured is client specific but it must be provided. Table 6 illustrates metadata traceability by providing details on how the data from the source data set can be converted to a different unit, or a BQL value can be converted to a numeric value.

PARAMCD	DATASET	VARIABLE	ANALYSIS_ALGORITHM	ORIGIN
DBILISI	ADLB	AVAL	If LBSTRESC contains '<' then AVAL is set to the numeric portion of LBSTRESC divided by 2. Else AVAL is set to LBSTRESC converted to a numeric value.	Derived
DBILISI	ADLB	PARAM	Set to "Direct Bilirubin (umol/L)"	Assigned
DBILICN	ADLB	AVAL	If LBSTRESC contains '<' then compute AVAL by taking the numeric portion of LBSTRESC and convert to SI units by multiplying by 0.05848, then take that value and divide by 2. Else AVAL is set to LBSTRESC converted to a numeric value multiplied by 0.05848.	Derived
DBILICN	ADLB	PARAM	Set to "Direct Bilirubin (mg/dL)"	Assigned

Table 6 Illustration of Metadata Traceability

BASELINE VALUES IN ADSL

It is always tempting to put as much information into ADSL as possible so that everything is in one place. However, it is not always a good idea to incorporate the baseline values or other values that are time-dependent into ADSL. Some people think that all the results on a table or figure needs to come from one data set. However, there is no rule that states all the results for an output must come from the same data set.

A reason for not adding time point dependent values:

- For most analysis data sets the baseline visit may not be the same record that was flagged in SDTM, thus the algorithm to determine baseline visits would need to be incorporated into ADSL
- The analysis visits may not align with the collection visit, thus the algorithm to determine analysis visits would need to be incorporated into ADSL
- Using other ADaM data sets to populate time-dependent values causes a circular reference.

Recommendations

- If there is no need for baseline values other than to summarize baseline characteristics (e.g., baseline weight, baseline height), then do not add to ADSL. It is fine for the output to reference more than one ADaM data set. Avoid the extra work.
- If the time-dependent values are needed for specific types of analyses (i.e., covariates, subgrouping), then they can be added to ADSL using one or two approaches
    - Base them on the SDTM domains and then do cross-checks with the ADaM data sets to ensure that the correct values were selected
    - Create a pre-ADSL data set that will only contain the necessary treatment information that needs to be merged into the non-ADSL ADaM data sets, and then use the non-ADSL ADaM data sets to create the final ADSL with the time-dependent values. The concept of a pre-ADSL will be introduced in the upcoming release of ADaM IG 1.2 which is expected to be out for public review by end of 2017.

ROWS VERSUS COLUMNS

The BDS is a flexible structure that allows for the addition of derived data. However, there are specific rules governing when a variable can be created and/or added. Per the ADaM IG:

> "The ADaM BDS structure contains a central set of columns (i.e., variables) that represent the data being analyzed. These variables include the value being analyzed (e.g., AVAL) and the description of the value being analyzed (e.g., PARAM). Other columns in the dataset provide more information about the value being analyzed (e.g., subject identification) or describe and trace the derivation of it (e.g., DTYPE) or support the analysis of it (e.g., treatment variables, covariates)."[2]

If a new column does not adhere to the rules outlined in the ADaM IG, which allow for the addition of a new column, then it should be added as a new row. A new row can represent either a new/conceptual time point or a new parameter.

For example, BASE and CHG have the same unit of AVAL on the same row. However if the analysis requires change from baseline to be analyzed in a different unit, then a new row should be created with PARAMCD and PARAM representing the new unit. The creation of new variables that represent different units for AVAL, BASE and CHG are not allowed.

Recommendations

- Determine if the derived data is a parameter-invariant function of AVAL and/or BASE on the same row and the function does not require a transformation of BASE. If these criteria are met, then the derived data can be added as a new column.
- If the criteria are not met, determine if the derived data is a function of one or more rows within the same parameter either for the purpose of imputing missing timepoints or creating a conceptual timepoint (i.e., creating a timepoint needed for analysis). If derived data is for the creation of an analysis timepoint then it can be added as a new row within the existing parameter.
- If the analysis calls for multiple baseline definitions, then there should be a corresponding set of rows for each baseline definition, and the variable BASETYPE is utilized to distinguish between the different definitions. Alternatively the data can be split so that each baseline definition is stored in its own data set.
- For all other scenarios, the derived data would be added as a new parameter or captured in a separate data set.
- Refer to section 4.2.1 of the ADaM IG for full details on the rules that govern the addition of a new column versus a new row.

PARAMTYP VERSUS DTYPE

People often confuse DTYPE and PARAMTYP. DTYPE is to indicate that a specific derivation was used to create that particular record. PARAMTYP is used to indicate a parameter did **not previously exist**. PARAMTYP can **only** take on the value of 'DERIVED'. DTYPE can be used in conjunction with PARAMTYP if a new record using the indicated algorithm needs to be populated.

Recommendations

- If using ADaM IG v1.0 or v1.1, then remember that PARAMTYP contains the string 'PARAM' to indicate parameter and that 'D' in DTYPE stands for derivation.
- If using ADaM IG v1.2, PARAMTYP has been deprecated and this should no longer be an issue.

Illustration

Table 7 illustrates the concept of DTYPE. An additional record was created to capture "special-case analysis value" for an existing parameter. The record imputes missing visits based on the derivation of last observation carried forward and therefore, DTYPE = 'LOCF' indicates this logic.

PARAM	PARAMCD	AVISIT	ABLFL	AVAL	BASE	CHG	DTYPE
Weight (lb)	WEIGHTLB	Baseline	Y	220	220		
Weight (lb)	WEIGHTLB	Week 12		207	220	-13	
Weight (lb)	WEIGHTLB	Week 24		207	220	-13	
Weight (lb)	**WEIGHTLB**	**Week 36**		**207**	**220**	**-13**	**LOCF**
Weight (lb)	WEIGHTLB	Week 48		202	220	-18	
Weight (lb)	WEIGHTLB	Week 52		209	220	-11	

Table 7 Illustration of DTYPE for Records that Did Not Exist for a Parameter

Building from the example in Table 7, the data needs to be converted to a different unit. The parameter did not previously exist in the data and therefore in Table 8 PARAMTYP is utilized to indicate that this is a new parameter.

PARAM	PARAMCD	AVISIT	ABLFL	AVAL	BASE	CHG	PARAMTYP
Weight (kg)	WEIGHTKG	Baseline	Y	99.8	99.8		**DERIVED**
Weight (kg)	WEIGHTKG	Week 12		93.9	99.8	-5.9	**DERIVED**

PARAM	PARAMCD	AVISIT	ABLFL	AVAL	BASE	CHG	PARAMTYP
Weight (kg)	WEIGHTKG	Week 24		93.9	99.8	-5.9	**DERIVED**
Weight (kg)	WEIGHTKG	Week 48		91.6	99.8	-8.2	**DERIVED**
Weight (kg)	WEIGHTKG	Week 52		94.8	99.8	-5	**DERIVED**

Table 8 Illustration of PARAMTYP for a Parameter that Did Not Exist

Since the Table 8 was building off of Table 7, it would be expected that all records in Table 7 would need to be accounted for, even the one where the missing visit was imputed. Table 9 illustrates the use of both DTYPE and PARAMTYP when a new parameter is created but a new record also needs to be created.

PARAM	PARAMCD	AVISIT	ABLFL	AVAL	BASE	CHG	DTYPE	PARAMTYP
Weight (kg)	WEIGHTKG	Baseline	Y	99.8	99.8			DERIVED
Weight (kg)	WEIGHTKG	Week 12		93.9	99.8	-5.9		DERIVED
Weight (kg)	WEIGHTKG	Week 24		93.9	99.8	-5.9		DERIVED
Weight (kg)	**WEIGHTKG**	**Week 36**		**93.9**	**99.8**	**-5.9**	**LOCF**	**DERIVED**
Weight (kg)	WEIGHTKG	Week 48		91.6	99.8	-8.2		DERIVED
Weight (kg)	WEIGHTKG	Week 52		94.8	99.8	-5		DERIVED

Table 9 Illustration of DTYPE and PARAMTYP on Same Record

FULL DESCRIPTION OF AVAL

People often use --TEST from the SDTM domain to populate PARAM, and use the corresponding qualifier variables in the SDTM domain to give a more accurate description of the analysis value. However, per the ADaM IG section 1.5.2 an analysis parameter (PARAM) is "a row identifier used to uniquely characterize a group of values that share a common definition…In contrast, SDTM –TEST column may need to be combined with qualifier columns such as --POS, --LOC, --SPEC, etc., in order to identify a group of related values."[2] For example, in LB domain LBTEST = 'GLUCOSE' does not tell us if the value is a Serum, Plasma or Urine Glucose. It does not indicate if the value was Fasting glucose nor does it specify what units the result is recorded. To fully describe the analysis value, PARAM would need to include the specimen (e.g., Serum, Plasma or Urine), the fasting status and the units to yield something like 'Serum Fasting Glucose (mg/dL)'.

Recommendations

- Any qualifier that might be relevant to analysis of the value should be included in the value of PARAM.
- PARAM should uniquely identify the contents of the analysis value.

POPULATING AVAL AND AVALC

In BDS, we have seen where individuals populate both AVAL and AVALC but there is not a 1-1 match within a PARAMCD. There is no requirement that they both be populated if they are in the data set. The only requirement in the ADaM IG is that either AVAL or AVALC be in the data set. Per the ADaM IG "on a given record, it is permissible for AVAL, AVALC, or both to be null."[2] However, if both variables are populated then there should be a 1-1 match within the parameter.

Recommendations

- If the result contains a below quantifiable limit (BQL) or above quantifiable limit (AQL) value and a formula is implemented to convert these values to a numeric value, then AVALC should not be populated with the BQL/AQL result. Instead --STRESC should be brought over to show data point traceability and AVALC left null so that AVAL can be populated using the formula. This allows the 1-1 match between AVAL and AVALC to be maintained.
- If the parameter is a quantitative value but the character version is needed, then --STRESC should be carried over from the corresponding SDTM domain and only AVAL should be populated.
- If the parameter is qualitative then AVALC should be populated and AVAL is typically left null. The only time AVAL should be populated for a qualitative value is when the values need to be ordered.

Illustration

For example, supposed that PARAMCD = URBC can have the following values specified in Table 10. However, these values are not ordered and therefore would not print as expected when the data is sorted. Instead the values

would print in the order indicated in Table 11. In order to get the data to print as expected, AVAL can be used to order AVALC as long as the 1-1 match constraint is adhered to, as illustrated in in Table 12.

AVALC
1-5
6-9
10-15
TNTC

Table 10 AVALC Unordered

AVALC (ordered)
1-5
10-15
6-9
TNTC

Table 11 AVALC Ordered Alphabetically

AVAL	AVALC
1	1-5
2	6-9
3	10-15
4	TNTC

Table 12 Using AVAL to Order AVALC

IMPUTATION OF MISSING OR PARTIAL DATES AND TIMES

It is common practice to impute missing or partial dates and/or times. However, when imputing dates/times, people tend to overlook the imputation flags. These date/time imputation flags indicate the level of imputation. Per the ADaM IG, "when a date or time is imputed, it is required that the variable containing the imputed value be accompanied by a date or time imputation flag variable."[2] Without the imputation flags it will be assumed that the value was a complete date/time.

Recommendations

- For pre-defined timing variables, the corresponding pre-defined imputation flags should be used.
- For date/time variables that are not pre-defined in the ADaM IG, date imputation flags should end in the variable fragment -DTF and time imputation flags should end in the variable fragment -TMF.
- The imputation flags should be set to the highest level of imputation even if a lower level was available.
- Refer to section 3.1.3 of the ADaM IG for full details on date and time imputation flag variables.

PARAMETER-LEVEL AND RECORD-LEVEL POPULATION FLAGS

There are three different types of population flags: subject-level, parameter-level and record-level. Anyone that has produced or used an ADSL data set is familiar with subject-level population flags. These flags indicate which population the subject is eligible for. Parameter-level and record-level population flags are less common and most of the time improperly used. Parameter-level population flags are used in BDS while record-level flags can be used in any non-ADSL structure.

Parameter-level flags (*PFL) identify a parameter that is typically included in the population analysis but is to be excluded for a particular type of analyses or for a particular subject within the population. Exclusion could be due to wanting to keep only parameters that align with the baseline record (e.g., subject has vitals taken and can be in different positions but you only want to keep the parameters that match the baseline position, so all other vital parameters not matching baseline are excluded for that subject). Record-level flags (*RFL) identify a specific record that would typically be eligible for assessment with the population but for some reason is being excluded. Exclusion could be due to the record occurring after a specific event.

Recommendations

- If within a BDS data set, a specific parameter needs to be excluded from a population due to some specific criteria, then a parameter-level population flag should be set.
- If within a non-ADSL data set and subject is eligible for a population based on the subject-level population flag but only specific records are to be excluded, then a record-level population flag should be set.

Illustration

For example, assume that the intent-to-treat analysis for the lab data set includes only parameters for subjects that were part of the ITT population and had a baseline and post-baseline assessment for a parameter. In Table 13 we show that subject 1001 is part of the ITT population but PARAMCD = 'TEST2' is not included in the ITT parameter-level population (ITTPFL) since it only had a baseline value. In addition subject 1003, PARAMCD = 'TEST1' is excluded since there was no corresponding baseline value.

USUBJID	ITTFL	PARAMCD	AVISIT	ADT	ABLFL	ITTPFL
1001	Y	TEST1	Screening	14NOV2015		Y
1001	Y	TEST1	Day 1	13DEC2015	Y	Y
1001	Y	TEST1	Week 4	18JAN2016		Y
1001	**Y**	**TEST2**	**Day 1**	**13DEC2015**	**Y**	
1002	Y	TEST1	Day 1	05MAR2016	Y	Y
1002	Y	TEST1	Week 4	08APR2016		Y
1003	**Y**	**TEST1**	**Week 4**	**15FEB2016**		
1003	**Y**	**TEST1**	**Week 8**	**10MAR2016**		
1004	Y	TEST1	Day 1	24MAY2016	Y	Y
1004	Y	TEST1	Week 4	19JUN2016		Y
1004	Y	TEST1	Week 8	26JUL2016		Y

Table 13 Illustration of Parameter-Level Population Flag

To illustrate the concept of record-level population flags, assume that the Statistical Analysis Plan (SAP) states that the Per Protocol analysis for the lab includes all records for a subject if the subject is part of the Per Protocol population and completed the study or if they discontinued the study for any reason other than discontinuation due to study drug. If subject discontinued due to study drug, then only records prior to discontinuation are included in the Per Protocol analysis. In Table 14, note that subject 1002 has both records for Per Protocol record-level flag set to null since the subject was not originally included in the Per Protocol population. However, for subject 1004 only the last record is excluded because the subject discontinued due to study drug and the assessment date is the same as the discontinuation date. All other records prior to discontinuation are included.

USUBJID	PPROTFL	EOSSTT	EOSDT	DCSREAS	PARAMCD	ADT	PPROTRFL
1001	Y	COMPLETED	15MAY2016		TEST1	14NOV2015	Y
1001	Y	COMPLETED	15MAY2016		TEST1	13DEC2015	Y
1001	Y	COMPLETED	15MAY2016		TEST1	18JAN2016	Y
1001	Y	COMPLETED	15MAY2016		TEST2	13DEC2015	Y
1002	**N**	**COMPLETED**	**24JUL2016**		**TEST1**	**05MAR2016**	
1002	**N**	**COMPLETED**	**24JUL2016**		**TEST1**	**08APR2016**	
1003	Y	DISCONTINUED	10MAR2016	ADVERSE EVENT	TEST1	15FEB2016	Y
1003	Y	DISCONTINUED	10MAR2016	ADVERSE EVENT	TEST1	10MAR2016	Y
1004	Y	DISCONTINUED	26JUL2016	DISCONTINUED STUDY DRUG	TEST1	24MAY2016	Y
1004	Y	DISCONTINUED	26JUL2016	DISCONTINUED STUDY DRUG	TEST1	19JUN2016	Y
1004	**Y**	**DISCONTINUED**	**26JUL2016**	**DISCONTINUED STUDY DRUG**	**TEST1**	**26JUL2016**	

Table 14 Illustration of Record-Level Population Flag

MULTIPLE BASELINE DEFINITIONS

As mentioned previously if there is more than one baseline definition, then BASETYPE needs to be incorporated. There should be only one baseline record per parameter, per baseline definition, per subject. If analysis calls for multiple baseline definitions, then either split the data so that each data set contains a different baseline definition (data set name and label should clearly identify the specific baseline definition) or create a new set of records and use BASETYPE to identify the records for each baseline definition

Recommendations

- For small data sets, incorporate the extra set of rows for each baseline definition and populate BASETYPE accordingly on all rows that will be used for that baseline definition.
- Split the data based into different data sets if
  - The data set is large and may require splitting at a later date

- There are several different baseline definitions and keeping in the same data set would cause confusion

Illustration

Suppose that ECG data is captured in triplicate and that the change from baseline analysis is to be based off of the minimum value and the maximum value of the set of values. Since there are two baseline definitions, then a complete set of records for each baseline definition should be created and the ABLFL should be set to flag the record that meets the baseline definition specified in BASETYPE. BASE should then be populated with the value that corresponds to the record where ABLFL = 'Y' as illustrated in Table 15

USUBJID	PARAMCD	AVISIT	ADTM	ABLFL	AVAL	BASE	CHG	BASETYPE
ABC-001-001	QTCB	Baseline	25FEB2014:08:30:24	Y	449	449		MINIMUM
ABC-001-001	QTCB	Baseline	25FEB2014:08:31:07		474	449		MINIMUM
ABC-001-001	QTCB	Baseline	25FEB2014:08:31:41		477	449		MINIMUM
ABC-001-001	QTCB	Day 3	27FEB2014:09:13:55		457	449	8	MINIMUM
ABC-001-001	QTCB	Day 3	27FEB2014:09:14:28		469	449	20	MINIMUM
ABC-001-001	QTCB	Day 3	27FEB2014:09:14:55		456	449	7	MINIMUM
ABC-001-001	QTCB	Week 2	13MAR2014:09:29:35		500	449	51	MINIMUM
ABC-001-001	QTCB	Week 2	13MAR2014:09:30:04		495	449	46	MINIMUM
ABC-001-001	QTCB	Week 2	13MAR2014:09:30:45		480	449	31	MINIMUM
ABC-001-001	QTCB	Baseline	25FEB2014:08:30:24		449	477		MAXIMUM
ABC-001-001	QTCB	Baseline	25FEB2014:08:31:07		474	477		MAXIMUM
ABC-001-001	QTCB	Baseline	25FEB2014:08:31:41	Y	477	477		MAXIMUM
ABC-001-001	QTCB	Day 3	27FEB2014:09:13:55		457	477	-20	MAXIMUM
ABC-001-001	QTCB	Day 3	27FEB2014:09:14:28		469	477	-8	MAXIMUM
ABC-001-001	QTCB	Day 3	27FEB2014:09:14:55		456	477	-21	MAXIMUM
ABC-001-001	QTCB	Week 2	13MAR2014:09:29:35		500	477	23	MAXIMUM
ABC-001-001	QTCB	Week 2	13MAR2014:09:30:04		495	477	18	MAXIMUM
ABC-001-001	QTCB	Week 2	13MAR2014:09:30:45		480	477	3	MAXIMUM

Table 15 Illustration of Multiple Baseline Definitions

CATEGORY AND CRITERION VARIABLES

The category and criterion variables that are reserved for BDS typically lend to confusion as to which variable(s) should be used for specific scenarios. A brief description of the variables is outlined in Table 16.

CATEGORY/CRITERIA VARIABLE	DESCRIPTION	BASED ON:
AVALCATy	Categorizes the analysis value into mutually exclusive groups	AVAL
BASECATy	Categorizes the baseline value into mutually exclusive groups	BASE
(P)CHGCATy	Categorizes the analysis value (percent) change from baseline into mutually exclusive groups	(P)CHG
(M)CRITy	Flags a record indicating if it met a specific criterion. Criterion can be binary (CRITy/CRITyFL) or it can have multiple response (MCRITy/MCRITyML).	Any variable on the ***same*** row
ANLzzFL	Flag used to select a record for analysis when the existing variables are not sufficient to select the record based on specific criteria.	Any variable and/or any row

Table 16 BDS Category and Criterion Flag Variables

Recommendations

- If the categories are mutually exclusive and are based off of AVAL, BASE, CHG or PCHG, then the appropriate CATy variable should be used.
- If AVAL is associated with at **most one** category, then use CATy. The same would apply with BASE, CHG and PCHG.

- If AVAL is associated with more than one category, then use (M)CRITy. The same would apply with BASE, CHG and PCHG.
- If the criteria is based off other variables on the **same** row, then
  - If the criteria is binary (i.e., Y or N), then CRITy should be used.
  - If the criteria have multiple responses, then MCRITy should be used.
- If the criteria is based off of other rows, then
  - If trying to select a specific record because the variables available are not adequate to uniquely select a record for analysis, then use ANLzzFL
  - For other scenarios it would probably be best to create a new parameter that contains the criteria and use AVAL/AVALC to capture the result.

Illustration

For the ECG analyses, the table shells indicate that the tables are to analyze the data based on the following criteria.

- Output 1: The number of subjects per visit that fall within the each of the following categories:
  - QTCB <= 420
  - QTCB 420 - < 450
  - QTCB 450 - < 480
  - QTCB >=480
- Output 2: The number of subjects per visit that had QTCB > 420 and the percent change from baseline > 5%
- Output 3: The number of subjects per visit that had QTCB > 420 and the percent change from baseline
  - 5% - < 10%
  - 10 % - < 15%
  - 15% - < 20%
  - >= 20%
- Output 4: Produce summary statistics for the maximum value at each visit.

For Output 1, the groups are mutually exclusive and the grouping is determined using AVAL, therefore AVALCATy should be used to group the data per the specifications. For Output 2, since the criterion is based off of AVAL and PCHG, then neither AVALCATy nor PCHGCATy can be used since the criteria are not one or the other. Thus, the next option would be to use (M)CRITy. Since the response is either they met the criteria or they did not meet the criteria, then CRITy should be utilized. The third output builds upon the second output in that it takes the subjects that met the criteria for output 2 and allows for a break down into additional responses. Therefore, MCRITy should be used in this situation. The last output requires that only the maximum value be used for each visit. Since the existing set of variables do not readily identify the maximum value per visit, ANLzzFL is used to flag the appropriate record. Table 17 illustrates the use of all these variables.

Row	USUBJID	PARAMCD	ADT	AVISIT	AVAL	BASE	CHG	PCHG	ABLFL	AVALCAT1
1	ABC-001-001	QTCB	07NOV2006	BASELINE	404	404			Y	<= 420
2	ABC-001-001	QTCB	21NOV2006	WEEK 2	410	404	6	1.5		<= 420
3	ABC-001-001	QTCB	07DEC2006	WEEK 4	425	404	21	5.2		420 - < 450
4	ABC-001-001	QTCB	31DEC2006	WEEK 8	460	404	56	13.9		450 - < 480
5	ABC-001-001	QTCB	03JAN2007	WEEK 8	458	404	54	13.4		450 - < 480
6	ABC-001-002	QTCB	08NOV2006	BASELINE	350	350			Y	<= 420
7	ABC-001-002	QTCB	22NOV2006	WEEK 2	375	350	25	7.1		<= 420
8	ABC-001-002	QTCB	12DEC2006	WEEK 4	435	350	85	24.3		420 - < 450

Row	CRIT1	CRIT1FL	MCRIT1	MCRIT1ML	ANL01FL
1	QTcB > 420 and PCHG > 5%		Percent Change Increase		Y
2	QTcB > 420 and PCHG > 5%	N	Percent Change Increase		Y
3	QTcB > 420 and PCHG > 5%	N	Percent Change Increase		Y
4	QTcB > 420 and PCHG > 5%	Y	Percent Change Increase	10% - <15%	Y
5	QTcB > 420 and PCHG > 5%	Y	Percent Change Increase	10% - <15%	
6	QTcB > 420 and PCHG > 5%		Percent Change Increase		Y
7	QTcB > 420 and PCHG > 5%	N	Percent Change Increase		Y
8	QTcB > 420 and PCHG > 5%	Y	Percent Change Increase	>= 20%	Y

Table 17 Illustration of AVALCATy, (M)CRITy and ANLzzFL

The following two papers have more information on when to use these variables.

- Proper Parenting: A Guide in Using ADaM Flag/Criterion Variables and When to Create a Child Dataset [7]
- ADaM Grouping: Groups, Categories, and Criteria. Which Way Should I Go? [6]

HANDLING ADVERSE EVENTS IN A CROSS-OVER STUDY

For cross-over studies, it may be necessary to analyze the data by incidence of the study treatment(s) versus by event onset. That is we want to count every treatment that was encountered during the duration of the event. Some have handled this by creating a number of new variables to account for all the possible different periods. This approach can make the data very 'wide' and very confusing to review. In addition, it does not allow for a standardization of data structure.

In section 1.1 of the ADaM Occurrence Data Structure, there are certain circumstances when the number of records in the analysis data set would not match the number in SDTM, such as when "an adverse event or concomitant medication, spans several treatment periods and needs to be counted in each. Based on the analysis need, a separate row might be required for each treatment period spanned and analyzed"[5]

Recommendations

- Determine if the analysis is to count the number of events based off of the onset of the event (i.e., typical analysis) or count the number of events at each treatment encounter (i.e., non-typical analysis).
  - If producing a typical analysis, then the standard data structure with one row per event should be created.
  - If producing a non-typical analysis, then a row for each treatment period the event was still present should be created.
  - In some cases, both types of analyses are needed. The analysis data set can be set up to allow both the typical and non-typical analysis of the study data.[5]

Illustration

Assuming we have a non-typical analysis, the AE will be counted in each period in which it occurred. Based on Table 18 and Table 19 it noted that the AE occurred after the end of treatment 1 and therefore, there is no need to create a record for APERIOD = 1. However, the AE started after initiation of treatment 2 and thus the AE is considered treatment emergent in APERIOD = 2. But there is no end date for the AE so for a non-typical analysis; the AE would also be counted in any subsequent periods. Hence, a record would need to be created for APERIOD = 3 as illustrated in Table 20. Note that only the record where APERIOD = 2 is set with TRTEMFL = 'Y' since this is when the AE emerged and the ANL01FL = 'Y' will help to select the correct record for a typical analysis. Thus, the one data set can handle both a typical and non-typical analysis.

USUBJID	AESEQ	AEDECOD	AESTDTC	AEENDTC	AEENRF
ABC-001-001	10	NAUSEA	2014-10-03		ONGOING

Table 18 AE Domain for Illustration of Adverse Events in Cross-over Study

USUBJID	TRTSEQP	TR01SDT	TR01EDT	TR02SDT	TR02EDT	TR03SDT	TR03EDT
ABC-001-001	A-B-C	2014-08-01	2014-09-18	2014-09-30	2014-11-16	2014-12-01	2015-01-18

Table 19 ADSL for Illustration of Adverse Events in Cross-over Study

USUBJID	AESEQ	AEDECOD	AESTDTC	AEENDTC	TRTA	TRTEMFL	APERIOD	ANL01FL
ABC-001-001	10	NAUSEA	2014-10-03		B	Y	2	Y
ABC-001-001	10	NAUSEA	2014-10-03		C		3	

Table 20 ADAE for Illustration of Adverse Events in Cross-over Study

STANDARDIZED MEDDRA QUERIES VERSUS CUSTOMIZED QUERIES

Handling of safety data used in analysis consistently focuses on AEs. It is becoming common to analyze AEs that are related or indicate a specific medical condition. Using Standardized MedDRA Queries (SMQs) and Customized Queries (CQs) is one such way to group related AEs. SMQs are pre-defined groupings that help to define a specific area of interest. Within each of these SMQs there are special features that can be used to refine the group (i.e., scope, class, hierarchy).[4] CQs are lists that are study specific. Unfortunately, the misuse of these types of variables is frequent.

We have seen where individuals try to incorporate more than one SMQ within a set of SMQ variables or try to combine an SMQ and CQ into one set of variables. This can cause issues if an AE occurs in more than one SMQ and/or CQ. "More than one SMQ cannot be stored within or under the same SMQzzNAM, SMQzzCD, SMQzzSC variables in your data set."[8]

Recommendations

- Determine if the AEs of interest are part of a pre-defined list
  - If so, then use the SMQ variables along with their special features since the majority of the work has been done up-front and there is no need for special look-up tables and custom programming.[8]
  - If not, then a CQ should be created using one of the four methods outlined in "Standardized, Customized or Both? Defining and Implementing (MedDRA) Queries in ADaM Data Sets".
- Store only one type of SMQ or CQ within the variable set.
- At the compound, or therapeutic area level, it may be necessary to keep the SMQ and CQ variables consistent for future integration.[8]

Illustration

Only one type of SMQ or CQ should be captured within a variable set since an AE can occur in multiple groups. As illustrated in Table 21. If multiple SMQs (or CQs) were captured in one variable set, then when an AE occurs in multiple groups, it must be determined which one should be stored in the first variable set. Furthermore, if multiple SMQs (or CQs) are captured in one set, then it is possible for the information to be stored in any of the variable sets. For example, since row 2 did not have Hepatic disorders (SMQ), we would not want to store Pregnancy and neonatal topics (SMQ) in SMQ11NAM because this would then require searching both SMQ11NAM and SMQ20NAM to find all AEs that are associated with Pregnancy and neonatal topics (SMQ). Therefore, to avoid having to search multiple variables to find all AEs associated with one particular SMQ (CQ), each SMQ (CQ) should have its own set of variables.

Row	AEDECOD	SMQ11NAM	SMQ11CD	SMQ11SC
1	Acute fatty liver of pregnancy	Hepatic disorders (SMQ)	20000005	Narrow
2	Agitation neonatal			

Row	SMQ20NAM	SMQ20CD	SMQ20SC
1	Pregnancy and neonatal topics (SMQ)	20000185	Narrow
2	Pregnancy and neonatal topics (SMQ)	20000185	Narrow

Table 21 Illustration of Multiple SMQs

ORIGINAL OR PRIOR CODING VARIABLES

For integrated studies, the data being pooled may use different dictionary versions. However, the integrated data needs to be coded using the same dictionary version so that analyses can be performed using a consistent version. We typically see people discarding the original coding values and overwriting using one dictionary version across all the pooled studies. With this approach, the coding associated with the individual studies' analyses is lost.

Recommendations

- If re-coding is needed due to a change in dictionary version but there is a need to maintain the prior coding because a prior analysis was performed, then the original/prior coding variables (*ORGw) should be kept to capture the prior coding information so that the MedDRA or WHODrug coding variables can be updated with the new coding information.
- Refer to section 3.2.10 of the ADaM Occurrence Data Structure (OCCDS)[3] for more details on these original/prior coding variables.

Illustration

There are times when several interim analyses are scheduled through the course of a study. As the study progresses, the coding is updated based on the most current dictionary version. Since it is possible for each interim analysis to have a different coding dictionary version, it is important that we maintain the coding associated with each interim for future reference. For example assume that this is a 3 year study, there is an interim analysis after the first and second year, and the coding dictionary will be updated as new versions of the dictionary are released. Table 22 illustrates how the original coding from the interim analyses can be retained.

Row	AETERM	AEDECOD	AELLT	AELLTCD	AEPTCD
1	Mosquito bite	Infected bite	Infected bites	10021769	10076911
2	Spider bite right arm	Infected bite	Insect bite, nonvenomous of shoulder and upper arm, infected	10022412	10076911

Row	DECDORG1	LLTORG1	LLTNORG1	DECDORG2	LLTORG2	LLTNORG2
1	Localised infection	Infected insect bite	10057257	Localised infection	Infected insect bite	10057257
2	Infected insect bite	Insect bite of should and upper arm, nonvenomous, infected	10022405	Infected insect bite	Insect bite, nonvenomous of should and upper arm, infected	10022412

Table 22 Illustration of Original/Prior Coding Variables

Although the coding terms illustrated in Table 22 appear to be legitimate MedDRA terms, they are for illustration purpose only and do not reflect specific MedDRA dictionary versions.

OTHER STRUCTURE

Although a structure other than one of the pre-defined structures (i.e., OTHER) can be used to set up an ADaM data set, it should not be standard practice. The OTHER structure should be the last resort.

Recommendations

- Most of the time data needed for analysis can fit nicely into a BDS or OCCDS, so exhaust all possible avenues before resorting to OTHER.
- Consult with a ADaM subject matter expert prior to using this structure
- If absolutely necessary to use the OTHER structure, ensure that **all** fundamental principles[1] are adhered to:
  - Clear and unambiguous communication with the use of either data point and/or metadata traceability.
  - ADaM data sets include metadata.
  - ADaM data sets are analysis-ready.
  - ADaM data sets and the associated metadata are machine readable by common software tools.

CONCLUSION

These are only some of the issues we have encountered when working with various clients across different analysis needs, and are by no means a definitive list of issues that may be encountered when working with ADaM data sets and their specifications. Although there are a number of potential issues that can have you end up in a sinkhole, these can be avoided if you follow the ADaM IG and ADaM OCCDS guidelines and/or advice of your company ADaM expert(s). Also, keep in mind that ADaM data sets should be analysis-ready, with all variable derivations performed

in the data set programs instead of the table programs. This will help you adhere to the fundamental principles found in the ADaM model, ADaM IG, and other supported guidelines, keep you ahead of the game and avoid timely and costly sinkholes.

REFERENCES

[1] CDISC Analysis Data Model Team. (2009). *Analysis Data Model (ADaM) v2.1.* http://www.cdisc.org/adam

[2] CDISC Analysis Data Model Team. (2016). *Analysis Data Model (ADaM) Implementation Guide version 1.1* http://www.cdisc.org/adam

[3] CDISC Analysis Data Model Team. (2016). *ADaM Occurrence Data Structure (OCCDS) (Version 1.0)* http://www.cdisc.org/adam

[4] ICH. (2016). *Introductory Guide for Standarised MedDRA Queries (SMQs) Version 19.1, MSSO-DI-6226-19.1.0, 2016, ICH*, http://www.meddra.org/sites/default/files/guidance/file/smq_intguide_19_1_english.pdf

[5] Miller, K., & Watson, R. (2015*).* Considerations in ADaM Occurrence Data: Handling Crossover Records for Non-Typical Analysis http://www.lexjansen.com/pharmasug/2015/DS/PharmaSUG-2015-DS06.pdf

[6] Shostak, J. (2017). ADaM Grouping: Groups, Categories, and Criteria. Which Way Should I Go? http://www.lexjansen.com/pharmasug/2017/DS/PharmaSUG-2017-DS17.pdf

[7] Watson, R., Miller, K., & Slagle, P. (2015). Proper Parenting: A Guide in Using ADaM Flag/Criterion Variables and When to Create a Child Dataset http://www.lexjansen.com/pharmasug/2015/DS/PharmaSUG-2015-DS08.pdf

[8] Watson, R., & Miller. K. (2016). Standardized, Customized or Both? Defining and Implementing (MedDRA) Queries in ADaM Data Sets http://www.lexjansen.com/pharmasug/2017/DS/PharmaSUG-2017-DS19.pdf

ACKNOWLEDGMENTS

We would like to thank Nancy Brucken for her thorough review and sanity check on some of these issues we have encountered.

RECOMMENDED READING

- *Analysis Data Model (ADaM) Implementation Guide version 1.1* http://www.cdisc.org/adam
- *ADaM Occurrence Data Structure (OCCDS) (Version 1.0)* http://www.cdisc.org/adam
- *Introductory Guide for Standarised MedDRA Queries (SMQs) Version 19.1, MSSO-DI-6226-19.1.0, 2016, ICH*, http://www.meddra.org/sites/default/files/guidance/file/smq_intguide_19_1_english.pdf

CONTACT INFORMATION

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

Richann Watson
Experis
richann.watson@experis.com

Karl Miller
inVentiv Health
karl.miller@inventivhealth.com