PharmaSUG 2025 - Paper SI-084

Approaches to Developing Multiple Imputation ADaM Dataset

Kang Xie, AbbVie Inc. Abbott Park, IL

ABSTRACT

Multiple Imputation (MI) is a statistical technique used to handle missing data by generating multiple complete datasets. These datasets are created by imputing missing values based on observed data, and each dataset is analyzed separately. The results from these analyses are then combined to provide a final inference.

This paper describes three examples of multiple imputation ADaM datasets generated by incorporating the imputed values into the ADaM dataset structure, maintaining the integrity of the analysis-ready format. Each example corresponds to a distinct multiple imputation method. The MI ADaM datasets include: 1) variables used in the imputation process and variables to identify the multiple imputation method, ensuring clear traceability, and 2) concatenated imputed values derived from the multiple imputation process.

Furthermore, this paper introduces the use of the PARQTYP and PARQUAL variables in the multiple imputation ADaM datasets. It illustrates how these variables apply to the analysis of multiple imputations, supporting decision-making regarding the standard criteria for PARQTYP and PARQUAL in ADaM.

INTRODUCTION

This paper describes the approaches to developing the MI ADaM dataset, illustrated through three examples. The MI ADaM dataset is used for the analysis where missing data is handled through multiple imputation methods.

The three examples are:

- **Example 1**: Multiple imputation using the two-step PROC MI with boundaries method, applied to all parameters in an MI ADaM dataset.
- **Example** 2: Multiple imputation using the two-step PROC MI method, where some parameters have boundaries applied, while others do not.
- **Example 3**: Multiple imputation using the one-step PROC MI with boundaries method, applied to all parameters in the MI ADaM dataset, where each parameter has two values, each value provided by an independent reviewer.

Additionally, this paper illustrates the use of PARQTYP and PARQUAL in multiple imputation ADaM dataset, which provides a scenario to support the discussion and decision-making regarding PARQTYP and PARQUAL in ADaM.

The views expressed in this paper are the authors' own views and interpretations of CDSIC standards, and do not represent the views of AbbVie Inc.

DETAILS OF MULTIPLE IMPUTATION (MI) ADAM DATASET DESIGN

EXAMPLE 1: MULTIPLE IMPUTATION USING THE TWO-STEP PROC MI WITH BOUNDARIES METHOD, APPLIED TO ALL PARAMETERS IN AN MI ADAM DATASET

Example 1 shows the MI ADaM dataset design for the output from the multiple imputation with boundaries using the two-step PROC MI method, applied to all parameters.

In this paper, the two-step PROC MI method is defined as the following: The first step utilizes the SAS PROC MI procedure with the MCMC method and the IMPUTE=MONOTONE option to create a monotone missing data pattern. The second step applies the PROC MI procedure with the MONOTONE REG method to impute the remaining missing values from Step 1.

Figure 1 displays a table that shows an example of missing data. Subject 102 has no missing value for each visit. Subject 101 has missing values in week 8 and week 12 for each of the 3 parameters (abscess count, inflammatory nodule count, and draining fistula count). These missing values are imputed through the two-step multiple imputation process (see Figure 2) and the imputed values are presented in multiple imputation ADaM Dataset shown in Figure 3.

Figure 1: Example of Observed Data in the Raw Dataset

SUBJID	IMPUTE	PARAM	PARAMCD	AVISIT	AVISITN	AVAL
101	OBS	Abscess Count	LCABT	Week 0	0	13
101	OBS	Abscess Count	LCABT	Week 2	2	14
101	OBS	Abscess Count	LCABT	Week 4	4	19
101	OBS	Inflammatory Nodule Count	LCINT	Week 0	0	13
101	OBS	Inflammatory Nodule Count	LCINT	Week 2	2	17
101	OBS	Inflammatory Nodule Count	LCINT	Week 4	4	25
101	OBS	Draining Fistula Count	LCDFT	Week 0	0	13
101	OBS	Draining Fistula Count	LCDFT	Week 2	2	15
101	OBS	Draining Fistula Count	LCDFT	Week 4	4	17
102	OBS	Abscess Count	LCABT	Week 0	0	9
102	OBS	Abscess Count	LCABT	Week 2	2	11
102	OBS	Abscess Count	LCABT	Week 4	4	9
102	OBS	Abscess Count	LCABT	Week 8	8	9
102	OBS	Abscess Count	LCABT	Week 12	12	13

Figure 2 presents the example SAS code for performing the two-step PROC MI with boundaries method.

Figure 2: Example SAS code for Two-Step PROC MI with Boundaries Method

```
proc mi data=lcabt out=lcabt_mono nimpute=20 seed=12345
  round= . . 1 1 1 1 1
  min= . . &minwk0 &minwk2 &minwk4 vminwk8 &minwk12
  max= . . &maxwk0 &maxwk2 &maxwk4 &maxwk8 &maxwk12;
  mcmc impute=monotone;
  var smokern trt01pn wk0 wk2 wk4 wk8 wk12;
run;
```

```
proc mi data=lcabt_mono out=lcabt_full  nimpute=1 seed=12345
round= . . 1 1 1 1 1
min= . . &minwk0 &minwk2 &minwk4 &minwk8 &minwk12
max= . . &maxwk0 &maxwk2 &maxwk4 &maxwk8 &maxwk12;
  class smokern trt01pn;
  var   smokern trt01pn wk0 wk2 wk4 wk8 wk12;
  monotone reg( wk2  = smokern trt01pn wk0);
  monotone reg( wk4  = smokern trt01pn wk0 wk2);
  monotone reg( wk8  = smokern trt01pn wk0 wk2 wk4);
  monotone reg( wk12 = smokern trt01pn wk0 wk2 wk4 wk8);
  by _imputation_;
run;
```

Figure 3 shows the key variables in the MI ADaM dataset that contains the output from the two-step multiple imputation process with boundary constraints applied. In this example, all parameters are applied with this imputation method. The "Row" is not a variable but is included for explanatory purposes in this paper.

- **IMPUNO (#1)** represents the imputation number, which identifies each imputed dataset generated through this multiple imputation method, The values range from 1 to 20, as defined by NIMPUTE=20 in #1 of Figure 2 above. In this MI ADaM dataset, only three sets of imputed datasets are shown (IMPUNO = 1, 2, and 3).
- **AVAL (#2)** contains the imputed values. For example, subject 101 has missing values for Weeks 8 and 12 (see Figure 1 above), which are imputed and the imputed values are stored in AVAL (#2) in rows 4 and 5 for the IMPUNO=1 dataset, and in the corresponding rows for the IMPUNO=2 and IMPUNO=3 datasets.
- **DTYPE (#3)** identifies the derivation type – "MI" – indicating that the missing values are imputed from multiple imputation method.
- The group of variables in **#4** represents the **classification covariates** used in the imputation model (refer to #2 in Figure 2 above). These covariates are included in the dataset to track the factors considered during the imputation process.
- The group of variables in **#5** are the **minimum and maximum variables** for each visit at the parameter level. The values of these variables are derived from the non-missing observed data in the source dataset in Figure 1. For example, minwk0 is the minimum abscess count number among all the observed data of Week 0 for a parameter, and maxwk0 is the maximum abscess count number among all the observed data of Week 0 for a parameter. These minimum and maximum values are used in the imputation model as the boundary constraints (refer to #3 in Figure 2 above), to ensure that the imputed values remain within the required range.

Figure 3: MI ADaM Dataset Example for Output from Two-Step PROC MI with Boundaries Method

ROW	USUBJID	PARAM	PARAMCD	IMPUNO	AVISIT	AVISITN	AVAL	DTYPE	SMOKERN	TRT01PN	MINWK0	MAXWK0	MINWK2	MAXWK2	MINWK4	MAXWK4	MINWK8	MAXWK8	MINWK12	MAXWK12
1	101	Abscess Count	LCABT	1	WK0	0	13		2	1	0	16	0	17	0	19	0	20	0	21
2	101	Abscess Count	LCABT	1	WK2	2	14		2	1	0	16	0	17	0	19	0	20	0	21
3	101	Abscess Count	LCABT	1	WK4	4	19		2	1	0	16	0	17	0	19	0	20	0	21
4	101	Abscess Count	LCABT	1	WK8	8	15	MI	2	1	0	16	0	17	0	19	0	20	0	21
5	101	Abscess Count	LCABT	1	WK12	12	13	MI	2	1	0	16	0	17	0	19	0	20	0	21
6	101	Abscess Count	LCABT	2	WK0	0	13		2	1	0	16	0	17	0	19	0	20	0	21
7	101	Abscess Count	LCABT	2	WK2	2	14		2	1	0	16	0	17	0	19	0	20	0	21
8	101	Abscess Count	LCABT	2	WK4	4	19		2	1	0	16	0	17	0	19	0	20	0	21
9	101	Abscess Count	LCABT	2	WK8	8	18	MI	2	1	0	16	0	17	0	19	0	20	0	21
10	101	Abscess Count	LCABT	2	WK12	12	21	MI	2	1	0	16	0	17	0	19	0	20	0	21
11	101	Abscess Count	LCABT	3	WK0	0	13		2	1	0	16	0	17	0	19	0	20	0	21
12	101	Abscess Count	LCABT	3	WK2	2	14		2	1	0	16	0	17	0	19	0	20	0	21
13	101	Abscess Count	LCABT	3	WK4	4	19		2	1	0	16	0	17	0	19	0	20	0	21
14	101	Abscess Count	LCABT	3	WK8	8	14	MI	2	1	0	16	0	17	0	19	0	20	0	21
15	101	Abscess Count	LCABT	3	WK12	12	17	MI	2	1	0	16	0	17	0	19	0	20	0	21
16	101	Draining Fistula Count	LCDFT	1	WK0	0	13		2	1	0	20	0	21	0	23	0	24	0	25
17	101	Draining Fistula Count	LCDFT	1	WK2	2	15		2	1	0	20	0	21	0	23	0	24	0	25
18	101	Draining Fistula Count	LCDFT	1	WK4	4	17		2	1	0	20	0	21	0	23	0	24	0	25
19	101	Draining Fistula Count	LCDFT	1	WK8	8	15	MI	2	1	0	20	0	21	0	23	0	24	0	25
20	101	Draining Fistula Count	LCDFT	1	WK12	12	13	MI	2	1	0	20	0	21	0	23	0	24	0	25
21	101	Draining Fistula Count	LCDFT	2	WK0	0	13		2	1	0	20	0	21	0	23	0	24	0	25
22	101	Draining Fistula Count	LCDFT	2	WK2	2	15		2	1	0	20	0	21	0	23	0	24	0	25
23	101	Draining Fistula Count	LCDFT	2	WK4	4	17		2	1	0	20	0	21	0	23	0	24	0	25
24	101	Draining Fistula Count	LCDFT	2	WK8	8	18	MI	2	1	0	20	0	21	0	23	0	24	0	25
25	101	Draining Fistula Count	LCDFT	2	WK12	12	21	MI	2	1	0	20	0	21	0	23	0	24	0	25
26	101	Draining Fistula Count	LCDFT	3	WK0	0	13		2	1	0	20	0	21	0	23	0	24	0	25
27	101	Draining Fistula Count	LCDFT	3	WK2	2	15		2	1	0	20	0	21	0	23	0	24	0	25
28	101	Draining Fistula Count	LCDFT	3	WK4	4	17		2	1	0	20	0	21	0	23	0	24	0	25
29	101	Draining Fistula Count	LCDFT	3	WK8	8	14	MI	2	1	0	20	0	21	0	23	0	24	0	25
30	101	Draining Fistula Count	LCDFT	3	WK12	12	17	MI	2	1	0	20	0	21	0	23	0	24	0	25

EXAMPLE 2: MULTIPLE IMPUTATION USING THE TWO-STEP PROC MI METHOD, WHERE SOME PARAMETERS HAVE BOUNDARIES APPLIED, WHILE OTHERS DO NOT

Example 2 illustrates the design of the MI ADaM dataset based on the output from the two-step multiple imputation process where boundaries model works for some parameters and does not work for others. In this example, both the boundaries and un-boundaries models are applied to each parameter. If the boundaries model works for a parameter, the imputed values from that model are used to fill in the missing data. If the boundaries model is not effective, the imputed values from un-boundaries model are used instead. As a result, each parameter is imputed either from the boundaries model or from the un-boundaries model, but not both. When imputed through the boundaries model, the dataset includes the minimum and maximum variables for each visit. However, when imputed through the un-boundaries model, these minimum and maximum variables for each visit are not included.

Figure 4 presents an example SAS code for the second step PROC MI method, showing two imputation models - one with boundaries and another without boundaries.

Figure 4: Example SAS code for the 2nd Step PROC MI Method with and without Boundary Constrains Applied

Imputation with boundaries

```
proc mi data=lcabt_mono out=lcabt_full  nimpute=1 seed=12345
  round= . . 1 1 1 1 1
  min= . . &minwk0 &minwk2 &minwk4 &minwk8 &minwk12
  max= . . &maxwk0 &maxwk2 &maxwk4 &maxwk8 &maxwk12;
  class smokern trt01pn;
  var   smokern trt01pn wk0 wk2 wk4 wk8 wk12;
  monotone reg( wk2  = smokern trt01pn wk0);
  monotone reg( wk4  = smokern trt01pn wk0 wk2);
  monotone reg( wk8  = smokern trt01pn wk0 wk2 wk4);
  monotone reg( wk12 = smokern trt01pn wk0 wk2 wk4 wk8);
  by _imputation_;
run;
```

Imputation without boundaries

```
proc mi data=lcabt_mono out=lcabt_full  nimpute=1 seed=12345
  round= . . 1 1 1 1 1
  min= . . . . . . .
  max= . . . . . . .;
  class smokern trt01pn;
  var   smokern trt01pn wk0 wk2 wk4 wk8 wk12;
  monotone reg( wk2  = smokern trt01pn wk0);
  monotone reg( wk4  = smokern trt01pn wk0 wk2);
  monotone reg( wk8  = smokern trt01pn wk0 wk2 wk4);
  monotone reg( wk12 = smokern trt01pn wk0 wk2 wk4 wk8);
  by _imputation_;
run;
```

Figure 5 highlights the key variables in the MI ADaM dataset designed for this example. In addition to the variables from Example 1, this dataset introduces a new variable, PARCAT1 (Parameter Category 1). If a parameter uses imputation with boundaries, "Two-Step PROC MI with Bound" is assigned to PARCAT1 to indicate that boundary constraints are applied for that parameter. If no boundary constraints are applied during the imputation process for a parameter, "Two-Step PROC MI without bound" is assigned to PARCAT1 instead. For example, subject 101 has PARCAT1 = "Two-Step PROC MI with bound" for the

“Abscess Count” parameter and "Two-Step PROC MI without bound" for another parameter - “Inflammatory Nodule Count”.

Figure 5: MI ADAM Dataset Example for Output from Two-Step PROC MI Method with and without Boundary Constraints Applied to Multiple Parameters

ROW	USUBJID	PARAM	PARAMCD	IMPUNO	AVISIT	AVISITN	AVAL	DTYPE	SMOKERN	TRT01PN	PARCAT1	MINWK0	MAXWK0	......	MINWK12	MAXWK12
1	101	Abscess Count	LCABT	1	WK0	0	13		2	1	Two-Step PROC MI with Bound	0	16	......	0	21
2	101	Abscess Count	LCABT	1	WK2	2	14		2	1	Two-Step PROC MI with Bound	0	16	......	0	21
3	101	Abscess Count	LCABT	1	WK4	4	19		2	1	Two-Step PROC MI with Bound	0	16	......	0	21
4	101	Abscess Count	LCABT	1	WK8	8	15	MI	2	1	Two-Step PROC MI with Bound	0	16	......	0	21
5	101	Abscess Count	LCABT	1	WK12	12	13	MI	2	1	Two-Step PROC MI with Bound	0	16	......	0	21
6	101	Abscess Count	LCABT	2	WK0	0	13		2	1	Two-Step PROC MI with Bound	0	16	......	0	21
7	101	Abscess Count	LCABT	2	WK2	2	14		2	1	Two-Step PROC MI with Bound	0	16	......	0	21
8	101	Abscess Count	LCABT	2	WK4	4	19		2	1	Two-Step PROC MI with Bound	0	16	......	0	21
9	101	Abscess Count	LCABT	2	WK8	8	18	MI	2	1	Two-Step PROC MI with Bound	0	16	......	0	21
10	101	Abscess Count	LCABT	2	WK12	12	21	MI	2	1	Two-Step PROC MI with Bound	0	16	......	0	21
11	101	Abscess Count	LCABT	3	WK0	0	13		2	1	Two-Step PROC MI with Bound	0	16	......	0	21
12	101	Abscess Count	LCABT	3	WK2	2	14		2	1	Two-Step PROC MI with Bound	0	16	......	0	21
13	101	Abscess Count	LCABT	3	WK4	4	19		2	1	Two-Step PROC MI with Bound	0	16	......	0	21
14	101	Abscess Count	LCABT	3	WK8	8	14	MI	2	1	Two-Step PROC MI with Bound	0	16	......	0	21
15	101	Abscess Count	LCABT	3	WK12	12	17	MI	2	1	Two-Step PROC MI with Bound	0	16	......	0	21
16	101	Inflammatory Nodule Count	LCINT	1	WK0	0	13		2	1	Two-Step PROC MI without bound					
17	101	Inflammatory Nodule Count	LCINT	1	WK2	2	17		2	1	Two-Step PROC MI without bound					
18	101	Inflammatory Nodule Count	LCINT	1	WK4	4	25		2	1	Two-Step PROC MI without bound					
19	101	Inflammatory Nodule Count	LCINT	1	WK8	8	18	MI	2	1	Two-Step PROC MI without bound					
20	101	Inflammatory Nodule Count	LCINT	1	WK12	12	25	MI	2	1	Two-Step PROC MI without bound					
21	101	Inflammatory Nodule Count	LCINT	2	WK0	0	13		2	1	Two-Step PROC MI without bound					
22	101	Inflammatory Nodule Count	LCINT	2	WK2	2	17		2	1	Two-Step PROC MI without bound					
23	101	Inflammatory Nodule Count	LCINT	2	WK4	4	25		2	1	Two-Step PROC MI without bound					
24	101	Inflammatory Nodule Count	LCINT	2	WK8	8	29	MI	2	1	Two-Step PROC MI without bound					
25	101	Inflammatory Nodule Count	LCINT	2	WK12	12	25	MI	2	1	Two-Step PROC MI without bound					
26	101	Inflammatory Nodule Count	LCINT	3	WK0	0	13		2	1	Two-Step PROC MI without bound					
27	101	Inflammatory Nodule Count	LCINT	3	WK2	2	17		2	1	Two-Step PROC MI without bound					
28	101	Inflammatory Nodule Count	LCINT	3	WK4	4	25		2	1	Two-Step PROC MI without bound					
29	101	Inflammatory Nodule Count	LCINT	3	WK8	8	18	MI	2	1	Two-Step PROC MI without bound					
30	101	Inflammatory Nodule Count	LCINT	3	WK12	12	15	MI	2	1	Two-Step PROC MI without bound					

EXAMPLE 3: MULTIPLE IMPUTATION USING THE ONE-STEP PROC MI WITH BOUNDARIES METHOD, APPLIED TO ALL PARAMETERS IN THE MI ADAM DATASET, WHERE EACH PARAMETER HAS TWO VALUES, EACH VALUE PROVIDED BY AN INDEPENDENT REVIEWER

Example 3 illustrates the design of the MI ADaM dataset based on the output from the one-step multiple imputation process. In this case, each parameter has two values for each visit, each value provided by an independent reviewer.

In this paper, the one-step PROC MI with boundaries method is defined as a single PROC MI procedure in which the MCMC method with the IMPUTE=FULL option is used to impute the missing data.

Figure 6 presents an example of SAS code for the one-step multiple imputation method. The code includes the "by reviewid" statement (#1), which will generate two distinct groups of observations: one for the data from reviewer 1 and another for the data from reviewer 2. Additionally, the MCMC method with the default option IMPUTE = FULL (#2) is applied, indicating that a full-data imputation will be performed for all missing values.

Figure 6: Example SAS code for One-Step PROC MI Procedure with Boundaries

```
proc mi data=jsn out=jsn_mi seed=12345 nimpute=20 round=1
  minimum= . . &minwk0 &minwk26 &minwk48 &minwk96
  maximum= . . &maxwk0 &maxwk26 &maxwk48 &maxwk96;
  by reviewid; 1
2 mcmc; *default is impute = full;
  var prbdmafl trt01p wk0 wk26 wk48 wk96;
run;
```

Figure 7 highlights the key variables in the MI ADaM dataset for this example. In this dataset, the new variables PARQTYP and PARQUAL are included, in addition to the variables from example 1. PARQTYP is used to specify that the parameter identifier type corresponds to Imaging Data, while PARQUAL is used to associate specific reviewer with the parameter (PARAM). For example, it identifies two reviewers: Reviewer 1 and Reviewer 2. The data in rows 1 to 12 correspond to Reviewer 1, while the data in rows 13 to 24 are for Reviewer 2.

Figure 7: MI ADAM Dataset Example for Output from One-Step PROC MI with Boundary Method Applied to Each Parameter with Two Values Each Value from an Independent Reviewer

ROW	USUBJID	PARQTYP	PARQUAL	PARAM	PARAMCD	IMPUNO	AVISIT	AVISITN	AVAL	DTYPE	PRBDMAFL	TRT01PN	MINWK0	MAXWK0	MINWK26	MAXWK26	MINWK48	MAXWK48	MINWK96	MAXWK96
1	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	1	BASELINE	0	21		Y	1	0	112	0	113	0	112	0	114
2	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	1	WEEK 26	26	20		Y	1	0	112	0	113	0	112	0	114
3	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	1	WEEK 48	48	21		Y	1	0	112	0	113	0	112	0	114
4	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	1	WEEK 96	96	22	MI	Y	1	0	112	0	113	0	112	0	114
5	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	2	BASELINE	0	21		Y	1	0	112	0	113	0	112	0	114
6	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	2	WEEK 26	26	20		Y	1	0	112	0	113	0	112	0	114
7	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	2	WEEK 48	48	21		Y	1	0	112	0	113	0	112	0	114
8	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	2	WEEK 96	96	20	MI	Y	1	0	112	0	113	0	112	0	114
9	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	3	BASELINE	0	21		Y	1	0	112	0	113	0	112	0	114
10	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	3	WEEK 26	26	20		Y	1	0	112	0	113	0	112	0	114
11	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	3	WEEK 48	48	21		Y	1	0	112	0	113	0	112	0	114
12	1001	Imaging Data	Reviewer 1	JSN Score	JSNSCORE	3	WEEK 96	96	19	MI	Y	1	0	113	0	113	0	110	0	113
13	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	1	BASELINE	0	22		Y	1	0	113	0	113	0	110	0	113
14	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	1	WEEK 26	26	22		Y	1	0	113	0	113	0	110	0	113
15	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	1	WEEK 48	48	19		Y	1	0	113	0	113	0	110	0	113
16	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	1	WEEK 96	96	22	MI	Y	1	0	113	0	113	0	110	0	113
17	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	2	BASELINE	0	22		Y	1	0	113	0	113	0	110	0	113
18	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	2	WEEK 26	26	22		Y	1	0	113	0	113	0	110	0	113
19	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	2	WEEK 48	48	19		Y	1	0	113	0	113	0	110	0	113
20	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	2	WEEK 96	96	21	MI	Y	1	0	113	0	113	0	110	0	113
21	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	3	BASELINE	0	22		Y	1	0	113	0	113	0	110	0	113
22	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	3	WEEK 26	26	22		Y	1	0	113	0	113	0	110	0	113
23	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	3	WEEK 48	48	19		Y	1	0	113	0	113	0	110	0	113
24	1001	Imaging Data	Reviewer 2	JSN Score	JSNSCORE	3	WEEK 96	96	22	MI	Y	1	0	113	0	113	0	110	0	113

PARQTYP and PARPUAL

PARQTYP is parameter qualifier type variable, while PARQUAL is parameter qualifier variable. Both are aimed at fully describing PARAM.

Currently, neither PARQTYP nor PARQUAL are included as BDS variables in ADaMIG 1.3. However, PARQTYP has been discussed in the CDSIC ADaM teams, and PARQUAL has been utilized in CDSIC ADaM examples and FDA documentation as shown in the four cases below.

Case 1: In the Prostate Cancer Therapeutic Area User Guide v1.0, PARQUAL is assigned as "INVESTIGATOR REVIEW" for investigator-based assessments, "INDEPENDENT REVIEW" for Independent Review assessments, and NULL for Overall Survival (OS) (refer to Figure 8).

Figure 8: PARQUAL Example Sourced from Prostate Cancer Therapeutic Area User Guide v1.0

adtte.xpt

Row	STUDYID	USUBJID	TRTP	PARAM	PARQUAL	PARAMCD	AVAL	CNSR	ADT
1	ABC-123	ABC-123-001	A	Progression-free survival (months)	INDEPENDENT REVIEW	PFS	10.8090349075975	0	15OCT2014
2	ABC-123	ABC-123-001	A	Progression-free survival (months)	INVESTIGATOR REVIEW	PFS	15.8090349075975	0	20OCT2014

Case 2: In the CDISC Breast Cancer Therapeutic Area User Guide v1.0, PARQUAL assigns "Investigator" for investigator-based assessments, "Central" for central imaging assessments, "Pathologic" for biopsy-based assessments, and "Protocol" for events influencing the assessment (refer to Figure 9).

Figure 9: PARQUAL Example Sourced from CDISC Breast Cancer Therapeutic Area User Guide v1.0

adtte.xpt **Proposed**

Row	STUDYID	USUBJID	*PARQUAL*	PARAMCD	AVAL	CNSR	SRCSEQ
1	ABC-123	ABC-123-001	INVESTIGATOR	PFS	87	0	11
2	ABC-123	ABC-123-001	CENTRAL	PFS	88	0	12
3	ABC-123	ABC-123-002	INVESTIGATOR	PFS	19	1	5
4	ABC-123	ABC-123-002	CENTRAL	PFS	20	1	6

Case 3: In the Online CDISC ADaM Oncology Examples, PARQUAL is used in the ADEXSUM data set to describe drug names (All, Drug Z) for summaries/evaluations of individual treatments and "All" for summaries/evaluations across all treatments (refer to Figure 10).

Figure 10: PARQUAL Example Sourced from Online CDISC ADaM Oncology Examples

ADEXSUM

Row	STUDYID	USUBJID	PARAMCD	PARAM	AEVLINT	PARQUAL	AVAL
1	ABC123	ABC123- 0201	TRTDURD	Treatment Duration Actual in Days	Overall	All	32
3	ABC123	ABC123- 0201	NADMIN	Nr of Actual Study Drug Administrations	Overall	All	4
4	ABC123	ABC123- 0201	NUMCYC	Number of Actual Cycles	Overall	All	4
5	ABC123	ABC123- 0201	NUMPCYC	Number of Planned Cycles	Overall	All	5
6	ABC123	ABC123- 0201	CUMPLDOS	Cumulative Planned Dose	Overall	Drug Z	42.5
7	ABC123	ABC123- 0201	CUMACDOS	Cumulative Actual Dose	Overall	Drug Z	30

Case 4: In Pilot OCE/OOD Standard Safety Data Requests v1.3, PARQUAL is applied in the ADEXSUM data set to define treatments. Equal to EXTRT for summaries/evaluations of individual treatments, or 'All' for summaries / evaluations across all treatments (refer to Figure 11).

Figure 11: PARQUAL Example Sourced from Pilot OCE/OOD Standard Safety Data Requests v1.3

ADEXSUM: Exposure Summary Analysis Dataset (adexsum.xpt)

Structure: One record per subject per parameter per analysis interval

ADEXSUM Variable Name	Variable Label	Type	Codelist/ Controlled Terms	CDISC Core (SDTM or ADaM)	OCE/OOD Core	CDISC Variable (ADaMIG v1.1 or SDTM v3.3 or OCE/OOD v1.3=FDA)	OCE/OOD Additional Information
PARQUAL	Parameter Qualifier	Char		N/A	Req	FDA	Description of the treatment summarized on each record. Equal to EXTRT for summaries/ evaluations of individual treatments, or 'All' for summaries / evaluations across all treatments.

These examples highlight how PARQUAL is applied in different contexts to specify the source of the parameter or to further describe the parameters, whether it is from investigator, central, or protocol-based assessments or specify which drugs are involved in the treatment summaries/evaluations.

In this paper, Example 3 illustrates the use of PARQTYP and PARQUAL in the multiple imputation ADaM dataset data set, which provides a case to support the discussion and decision-making regarding PARQTYP and PARQUAL in ADaM.

CONCLUSION

This paper describes the design of multiple imputation ADaM dataset through three examples.

The example multiple imputation ADaM dataset has the following key features:

1. It is structured according to the BDS format.

2. It includes a set of covariate variables to track the factors considered during the imputation process.

3. It contains a group of boundary constraints variables that define the ranges applied in the imputation process.

4. It uses the parameter category variable (PARCAT1) to classify the parameters according to different imputation methods. For instance, in Example 2, PARCAT1 is used to specify whether boundary constraints were applied to a parameter during the imputation process.

5. It utilizes the parameter identifier type variable (PARQTYP) to indicate that the parameter identifier type corresponds to Imaging Data, and the parameter identifier variable (PARQUAL) to link the reviewers (Reviewer 1 and Reviewer 2) to the parameters (PARAM). This paper provides a case to support the discussion and decision-making regarding PARQTYP and PARQUAL in ADaM.

REFERENCES

SAS Institute Inc. 2020. SAS/STAT® 15.2 User's Guide. Cary, NC: SAS Institute Inc.

ADaMIG v1.3, 2021-12-29. https://www.cdisc.org/standards/foundational/ADaM/ADaMig-v1-3

Prostate Cancer Therapeutic Area User Guide v1.0, 2017-07-10 .
https://www.cdisc.org/standards/therapeutic-areas/prostate-cancer/prostate-cancer-therapeutic-area-user-guide-v1-0

CDISC Breast Cancer Therapeutic Area User Guide v1.0, 2016-05-16.
https://www.cdisc.org/standards/therapeutic-areas/breast-cancer/breast-cancer-therapeutic-area-user-guide-v1-0

CDISC ADaM Oncology Examples, https://wiki.cdisc.org/pages/viewpage.action?pageId=73139566

Pilot OOD Safety Team Standard Data Requests, 2021-02.
https://www.fda.gov/search?s=Pilot+OCE%2FOOD+Standard+Safety+Data+Requests+v1.3

ACKNOWLEDGMENTS

The author also appreciates the review and comments from her manager Ramesh C Gali.

CONTACT INFORMATION

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

Kang Xie
Kang.xie@abbvie.com