PharmaSUG 2014 - Paper AD11

Dealing with Missing Data in Clinical Trials

Lucheng Shao, Ivantis, Inc, Irvine, CA

ABSTRACT

It seems inevitable to encounter missing data in clinical trials no matter how perfect the study was designated and how carefully Clinical Research Associates collected the data. However, having missing values in our original clinical database is not the end of the world for our SAS® programmers. The focus of this paper will be showing you how to deal with missing data in clinical trials, especially how to improve the way of representing different types of missing data so that the missing information can be taken advantage of in improving reports. This paper does not cover missing data mechanisms or imputation methods. It is intended for readers who are similar with SAS base but not with different types of missing data.

INTRODUCTION

As a clinical study starts moving forward we will have different types of missing data. In an ophthalmic device study, examples of missing data could be any of, but not limited to, the following:

- A missing stop date for certain medication on the medication CRF (missing reason: the subject was still on medication by the visit date).
- A missing Diurnal Intraocular Pressure (DIOP) value with the left eye (missing reason: the study eye was the right eye).
- A missing DIOP value with the study eye due to subject discontinuation (e.g., death, replacement of an implanted device), lost to follow up, visit outstanding, etc.
- A missing DIOP value due to the fact that the subject was not washed out. A DIOP value would only be recorded if the subject is not on medication or has been washed out by the visit.
- In a surgical postoperative questionnaire an answer may be missing because the subject refused to answer a sensitive question or the section was skipped because it was not applicable (e.g., Question 20 was set to be skipped if the subject answered 'No' to Question 19.)

Example 1 mimics data collected from an ophthalmic device clinical study. The dataset includes 4 subjects (111, 112, 113, 114), three of whom were randomized into the procedure.

- Subject 111 was lost to follow up some time after the 1 Day postoperative visit.
- Subject 112 hadn't got his 12 Month postoperative visit done by the date that the dataset was prepared.
- Subject 113 hadn't been washed out by the time he had the baseline visit thus his DIOP value at Baseline was missing.
- Subject 114 failed to pass the study inclusion criteria thus exited the study before making the baseline visit.

Example 1: A sample dataset from an ophthalmic device clinical study

Subject	OE	Trt_group	VisitDt	Visit_Name	DIOP_OD	DIOP_OS
111	OD	(Not assigned)	5/19/2011	Screening	19.6	21.5
112	OD	(Not assigned)	5/25/2011	Screening	26.7	24.9
113	OS	(Not assigned)	6/3/2011	Screening	17.2	18.8
114	OS	(Not assigned)	7/3/2011	Screening	15.4	14.7

111	OD	(Not assigned)	6/29/2011	Baseline	21.3	(Non Study Eye)
112	OD	(Not assigned)	6/30/2011	Baseline	34.6	(Non Study Eye)
113	OS	(Not assigned)	7/7/2011	Baseline	(Non Study Eye)	(Not Washed Out)
114	OS	(Exit)	(Exit)	Baseline	(Exit)	(Exit)
111	OD	1	7/1/2011	Postop_1 Day	18.2	(Non Study Eye)
112	OD	2	7/2/2011	Postop_1 Day	24.3	(Non Study Eye)
113	OS	2	7/9/2011	Postop_1 Day	(Non Study Eye)	20.8
114	OS	(Exit)	(Exit)	Postop_1 Day	(Exit)	(Exit)
111	OD	1	(Lost to follow up)	Postop_1 Month	(Lost to follow up)	(Non Study Eye)
112	OD	2	7/28/2011	Postop_1 Month	21.8	(Non Study Eye)
113	OS	2	8/6/2011	Postop_1 Month	(Non Study Eye)	18.1
114	OS	(Exit)	(Exit)	Postop_1 Month	(Exit)	(Exit)
111	OD	1	(Lost to follow up)	Postop_12 Month	(Lost to follow up)	(Non Study Eye)
112	OD	2	(Not Done)	Postop_12 Month	(Not Done)	(Non Study Eye)
113	OS	2	6/20/2012	Postop_12 Month	(Non Study Eye)	17.9
114	OS	(Exit)	(Exit)	Postop_12 Month	(Exit)	(Exit)

Note: In Example 1, the text within parentheses () indicates the reason for missing data points.
Notations: OE = Study Eye; OD = Right Eye; OS = Left Eye; Trt_group = Treatment Group; VisitDt = Visit Date; Visit_Name= the name of the visit; DIOP_OD = DIOP value of the right eye; DIOP_OS = DIOP value of the left eye.

In SAS a missing character is normally represented by a blank (‘ ‘) and a missing numeric value is represented by a period (‘.’). Since Example 1 includes only missing numeric values, the raw dataset in Example 1 will look like the following (Example 1.1) after being imported into SAS.

Example 1.1: SAS Output of Example 1 using PROC PRINT

Obs	Subject	OE	Trt_group	VisitDt	Visit_Name	DIOP_OD	DIOP_OS
1	111	OD	.	05/19/2011	Screening	19.6	21.5
2	112	OD	.	05/25/2011	Screening	26.7	24.9
3	113	OS	.	06/03/2011	Screening	17.2	18.8
4	114	OS	.	07/03/2011	Screening	15.4	14.7
5	111	OD	.	06/29/2011	Baseline	21.3	.
6	112	OD	.	06/30/2011	Baseline	34.6	.
7	113	OS	.	07/07/2011	Baseline	.	.

8	114	OS	.	.	Baseline	.	.
9	111	OD	1	07/01/2011	Postop_1 Day	18.2	.
10	112	OD	2	07/02/2011	Postop_1 Day	24.3	.
11	113	OS	2	07/09/2011	Postop_1 Day	.	20.8
12	114	OS	.	.	Postop_1 Day	.	.
13	111	OD	1	.	Postop_1 Month	.	.
14	112	OD	2	07/28/2011	Postop_1 Month	21.8	.
15	113	OS	2	08/06/2011	Postop_1 Month	.	18.1
16	114	OS	.	.	Postop_1 Month	.	.
17	111	OD	1	.	Postop_12 Month	.	.
18	112	OD	2	.	Postop_12 Month	.	.
19	113	OS	2	06/20/2012	Postop_12 Month	.	17.9
20	114	OS	.	.	Postop_12 Month	.	.

CAN SPECIAL NUMBERS BE A SOLUTION FOR MISSING NUMERIC VALUES?

This works well with date variables. Suppose that our task is to provide a subject listing that shows the number of medications that each subject was taking at each visit. One possible way to approach this (see Example 2) is to compare the visit date to the medication's start and stop date, and see if the visit date was within that range that is formed by start and stop date. In such a case, it would not be surprising to see missing values in the stop date variable since the subject could be still taking the medication.

Example 2: Missing Stop Date Example. This dataset is named 'meds' in SAS

Obs	PID	VisitDt	Visit_Name	Medication	StartDt	StopDt
1	111	05/19/2011	Screening	Meds_A	09/10/2010	05/20/2011
2	112	05/25/2011	Screening	Meds_B	03/13/2010	02/18/2011
3	113	06/03/2011	Screening	Meds_C	03/23/2010	.

Code 1 of Example 2 shows the common logic of counting medications. Suppose that when day 1 was recorded as a start date (or a stop date) in CRF, day 2 is the real day on which the medication was started (or stopped), then 'StartDt lt VisitDt le StopDt' serves as a reasonable decision criterion in the process of counting medications.

```
***Code 1 of Example 2;
data meds_2;
set meds;
if StartDt lt VisitDt le StopDt then meds_count=1;
else meds_count=0;
run;

proc print data=meds_2;
run;
```

Output Example 2.1: Output of Example 2 using Code 1

Obs	PID	VisitDt	Visit_Name	Medication	StartDt	StopDt	meds_count
1	111	05/19/2011	Screening	Meds_A	09/10/2010	05/20/2011	1
2	112	05/25/2011	Screening	Meds_B	03/13/2010	02/18/2011	0
3	113	06/03/2011	Screening	Meds_C	03/23/2010	.	0

In the output (Example 2.1), we can see that when the StartDt variable and the StopDt variable both have a non-missing value the program generates correct medication count. However, the medication number of subject 113 was mistakenly counted as 0. This is because SAS is not 'smart' enough to see the missing value as 'the subject was still on medication by the visit day'. We have to tell SAS about this information explicitly, one way is to assign a real but well-into-the-future date to the StopDt variable (see Example 2 Code 2).

```
***Code 2 of Example 2;
data meds_2;
set meds;
if StopDt=. then StopDt='31Dec2050'd;
if StartDt lt VisitDt le StopDt then meds_count=1;
else meds_count=0;
run;

proc print data=meds_2;
run;
```

Example 2.2: Output of Example 2 using Code 2

Obs	PID	VisitDt	Visit_Name	Medication	StartDt	StopDt	meds_count
1	111	05/19/2011	Screening	Meds_A	09/10/2010	05/20/2011	1
2	112	05/25/2011	Screening	Meds_B	03/13/2010	02/18/2011	0
3	113	06/03/2011	Screening	Meds_C	03/23/2010	12/31/2050	1

Converting a missing date into another proper but non-missing date becomes a perfect solution for problems that involve date range checks. Now the question becomes, can we generalize this solution to any missing numeric data problems? Unfortunately, the answer is 'No'. Consider a very simple example (see Example 3) in which we have three missing DIOP values at 12 Month Visit. The PROC MEANS in code 1 automatically excludes these missing values when calculating statistics (N, Mean, Std Dev, Minimum, Maximum), and it generates correct results.

If we converted those missing values into some special number such as '999' in code 2, the result becomes skewed and of course incorrect. A take home message here is to convert a missing date into a proper future date when necessary and leave any other type of missing numeric data as it is. We do have a better way of dealing with those missing numeric data, which will be introduced in the following part.

```
***Code 1 of Example 3;
data DIOP_case1;
input DIOP_Screening DIOP_Baseline DIOP_Month12 @@;
datalines;
19.0 23.8 18.2
24.5 25.7 .
30.8 32.4 .
26.7 27.9 23.2
29.4 29.9 .
;

proc means data=DIOP_case1;
run;
```

Example 3.1: Output of Example 3 using Code 1

Variable	N	Mean	StdDev	Minimum	Maximum
DIOP_Screening	5	26.0800000	4.6451050	19.0000000	30.8000000
DIOP_Baseline	5	27.9400000	3.3871817	23.8000000	32.4000000
DIOP_Month12	2	20.7000000	3.5355339	18.2000000	23.2000000

```
***Code 2 of Example 3;
data DIOP_case2;
input DIOP_Screening DIOP_Baseline DIOP_Month12 @@;
datalines;
19.0 23.8 18.2
24.5 25.7 999
30.8 32.4 999
26.7 27.9 23.2
29.4 29.9 999
;

proc means data=DIOP_case2;
run;
```

Example 3.2: Output of Example 3 using Code 2

Variable	N	Mean	StdDev	Minimum	Maximum
DIOP_Screening	5	26.0800000	4.6451050	19.0000000	30.8000000
DIOP_Baseline	5	27.9400000	3.3871817	23.8000000	32.4000000
DIOP_Month12	5	607.6800000	535.8398940	18.2000000	999.0000000

TAKING ADVANTANGE OF SPECIAL MISSING CHARACTERS MAKES OUR LIFE EASIER

In clinical studies a missing data point can be a result of several different reasons (eg., subject lost to follow up, visit not done, etc.). If we simply use a period ('.') to represent all kinds of missing numeric values, we will easily get lost in the missing forest and will never be able to tell which was the exact missing reason for each missing data point. Fortunately, we have several special characters in SAS that can help us assign and easily remember missing reasons. These special characters include '.A' to '.Z' and '._'. When blended with regular numeric values, the sort order is '._' <'.' (regular missing numeric values) < '.A' < '.Z' < negative values < 0 < positive values. Sometimes this order can be taken advantage of in keeping certain missing points as still missing while substituting other missing points with new non-missing data (eg., we can use selecting criterion such as variable > '._').

Now the raw data in Example 1 is modified by having all missing data represented by special missing characters (see Example 4). A new format called 'missing' is also generated to explain detailed missing reasons. An error message occurred when PROC PRINT was used to show the output.

Example 4: Modified Example 1 with all missing values represented by special characters.

Subject	OE	Trt_group	VisitDt	Visit_Name	DIOP_OD	DIOP_OS
111	OD	.R	5/19/2011	Screening	19.6	21.5
112	OD	.R	5/25/2011	Screening	26.7	24.9
113	OS	.R	6/3/2011	Screening	17.2	18.8
114	OS	.R	7/3/2011	Screening	15.4	14.7
111	OD	.R	6/29/2011	Baseline	21.3	.N

112	OD	.R	6/30/2011	Baseline	34.6	.N
113	OS	.R	7/7/2011	Baseline	.N	.W
114	OS	.E	.E	Baseline	.N	.E
111	OD	1	7/1/2011	Postop_1 Day	18.2	.N
112	OD	2	7/2/2011	Postop_1 Day	24.3	.N
113	OS	2	7/9/2011	Postop_1 Day	.N	20.8
114	OS	.E	.E	Postop_1 Day	.N	.E
111	OD	1	.L	Postop_1 Month	.L	.N
112	OD	2	7/28/2011	Postop_1 Month	21.8	.N
113	OS	2	8/6/2011	Postop_1 Month	.N	18.1
114	OS	.E	.E	Postop_1 Month	.N	.E
111	OD	1	.L	Postop_12 Month	.L	.N
112	OD	2	.D	Postop_12 Month	.	.N
113	OS	2	6/20/2012	Postop_12 Month	.N	17.9
114	OS	.E	.E	Postop_12 Month	.N	.E

```
***Example 4 Code 1;
proc format;
value missing .R = 'Not Randomized'
              .E = 'Exit'
              .L = 'Lost To Follow Up'
              .D = 'Not done'
              .N = 'Not Available'
              .W = 'Not Washed Out'
              ;
run;

proc print data=ex4;
format Trt_group VisitDt DIOP_OD DIOP_OS missing.;
run;

Log:
20603  proc print data=ex4;
20604  format Trt_group VisitDt DIOP_OD DIOP_OS missing.;
ERROR: You are trying to use the numeric format MISSING with the character variable
VisitDt in data set WORK.EX4.
20605  run;

NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE PRINT used (Total process time):
real time           0.00 seconds
cpu time            0.00 seconds
```

This happens because the date variable ‘VisitDt’, when imported from Excel, gets converted to a character variable; this happens because Excel would treat .E and the other special SAS missing values as character. If a data variable does not contain any character in it, it will be imported as a numeric variable; otherwise, it will be treated as a character variable. In Example 4, when Code 1 was applied, it was those special characters (‘.E’, ‘.L’ and ‘.D’) that caused ‘VisitDt’ to be treated as a character variable and thus triggered errors. The following code 2 converts the character variable ‘VisitDt’ into numeric by applying the INPUT function and generates the correct output.

```
***Example 4 Code 2;
proc format;
invalue missDates
        .E      = .E
        .L      = .L
        .D      = .D
        other   = [ANYDATE10.]
        ;

value missDates
         .E      = 'Exit'
         .L      = 'Lost to follow up'
         .D      = 'Not Done'
         other   = [yymmdd10.]
         ;
run;


data ex4_2(drop=dt);
retain Subject OE Trt_group VisitDt Visit_Name DIOP_OD DIOP_OS;
set ex4 (rename=(visitDT = dt));
visitDt = input(dt, missDates.);
run;

proc print data=ex4_2;
format visitDt missDates. Trt_group DIOP_OD DIOP_OS missing.;
run;
```

Output of Example 4 Code 2

Obs	Subject	OE	Trt_group	VisitDt	Visit_Name	DIOP_OD	DIOP_OS
1	111	OD	Not Randomized	2011-05-19	Screening	19.6	21.5
2	112	OD	Not Randomized	2011-05-25	Screening	26.7	24.9
3	113	OS	Not Randomized	2011-06-03	Screening	17.2	18.8
4	114	OS	Not Randomized	2011-07-03	Screening	15.4	14.7
5	111	OD	Not Randomized	2011-06-29	Baseline	21.3	Not Available
6	112	OD	Not Randomized	2011-06-30	Baseline	34.6	Not Available
7	113	OS	Not Randomized	2011-07-07	Baseline	Not Available	Not Washed Out
8	114	OS	Exit	Exit	Baseline	Not Available	Exit
9	111	OD	1	2011-07-01	Postop_1 Day	18.2	Not Available
10	112	OD	2	2011-07-02	Postop_1 Day	24.3	Not Available
11	113	OS	2	2011-07-09	Postop_1 Day	Not Available	20.8
12	114	OS	Exit	Exit	Postop_1 Day	Not Available	Exit
13	111	OD	1	Lost to	Postop_1	Lost To	Not

				follow up	Month	Follow Up	Available
14	112	OD	2	2011-07-28	Postop_1 Month	21.8	Not Available
15	113	OS	2	2011-08-06	Postop_1 Month	Not Available	18.1
16	114	OS	Exit	Exit	Postop_1 Month	Not Available	Exit
17	111	OD	1	Lost to follow up	Postop_12 Month	Lost To Follow Up	Not Available
18	112	OD	2	Not Done	Postop_12 Month	.	Not Available
19	113	OS	2	2012-06-20	Postop_12 Month	Not Available	17.9
20	114	OS	Exit	Exit	Postop_12 Month	Not Available	Exit

Taking advantage of special characters in representing missing values not only benefits us on clearly viewing the missing reasons but also provides a convenient way of retrieving and excluding a certain subset of missing values from the whole dataset (See Example 5.a and Example 5.b). The 'update' function is also very helpful when we would like to update certain missing values (See Example 5.c).

EXAMPLE 5.A SHOWS HOW TO RETRIEVE A SUBSET OF SUBJECTS WHO LOST TO FOLLOW UP FROM EX4_2:

```
data sub_LTFU;
set ex4_2;
if DIOP_OD = .L or DIOP_OS = .L;
run;

proc print data=sub_LTFU;
format visitDt missDates. Trt_group DIOP_OD DIOP_OS missing.;
run;
```

Output of Example 5.a

Obs	Subject	OE	Trt_group	VisitDt	Visit_Name	DIOP_OD	DIOP_OS
1	111	OD	1	Lost to follow up	Postop_1 Month	Lost To Follow Up	Not Available
2	111	OD	1	Lost to follow up	Postop_12 Month	Lost To Follow Up	Not Available

EXAMPLE 5.B SHOWS HOW TO RETRIEVE A SUBSET OF SUBJECTS WHO DID NOT EXIT THE STUDY.

```
data sub_N_e;
set ex4_2;
if DIOP_OD ^= .E and DIOP_OS ^= .E;
run;

proc print data=sub_N_e;
format visitDt missDates. Trt_group DIOP_OD DIOP_OS missing.;
run;
```

Output of Example 5.b

Obs	Subject	OE	Trt_group	VisitDt	Visit_Name	DIOP_OD	DIOP_OS
1	111	OD	Not Randomized	2011-05-19	Screening	19.6	21.5
2	112	OD	Not Randomized	2011-05-25	Screening	26.7	24.9
3	113	OS	Not Randomized	2011-06-03	Screening	17.2	18.8
4	114	OS	Not Randomized	2011-07-03	Screening	15.4	14.7
5	111	OD	Not Randomized	2011-06-29	Baseline	21.3	Not Available
6	112	OD	Not Randomized	2011-06-30	Baseline	34.6	Not Available
7	113	OS	Not Randomized	2011-07-07	Baseline	Not Available	Not Washed Out
8	111	OD	1	2011-07-01	Postop_1 Day	18.2	Not Available
9	112	OD	2	2011-07-02	Postop_1 Day	24.3	Not Available
10	113	OS	2	2011-07-09	Postop_1 Day	Not Available	20.8
11	111	OD	1	Lost to follow up	Postop_1 Month	Lost To Follow Up	Not Available
12	112	OD	2	2011-07-28	Postop_1 Month	21.8	Not Available
13	113	OS	2	2011-08-06	Postop_1 Month	Not Available	18.1
14	111	OD	1	Lost to follow up	Postop_12 Month	Lost To Follow Up	Not Available
15	112	OD	2	Not Done	Postop_12 Month	.	Not Available
16	113	OS	2	2012-06-20	Postop_12 Month	Not Available	17.9

EXAMPLE 5.C:

Suppose that after following up with the site, the CRA collected the 12 Month Visit Date and a DIOP value for subject 112. This information was saved in a new SAS dataset called 'new'.

```
proc print data=new;
run;
```

Output

Obs	Subject	VisitDt	Visit_Name	DIOP_OD	DIOP_OS
1	112	06/13/2012	Postop_12 Month	20.5	N

The following code shows how to update the missing 12 Month Visit in Example 4 (dataset 'ex4_2') with the information in the dataset 'new'. Note that a new variable 'VisitNo' was introduced in order to keep the order of visit names as 'Screening, Baseline, Postop_1 Day, Postop_1 Month and Postop_12 Month'; otherwise, if we use the variable 'Visit_Name' as the sort variable, visit names will be re-ordered alphabetically in the updated dataset.

```
proc format;
value visitfmt
      1 = 'Screening'
      2 = 'Baseline'
      3 = 'Postop_1 Day'
      4 = 'Postop_1 Month'
      5 = 'Postop_12 Month'
      ;

invalue visitfmt
       'Screening'       = 1
       'Baseline'        = 2
       'Postop_1 Day'    = 3
       'Postop_1 Month'  = 4
       'Postop_12 Month' = 5
       ;
run;

data new_o(drop=Visit_Name);
set new;
Visitno=5;
run;

proc sort data=new_o;
by Visitno Subject;
run;

data ex4_2_o(drop=Visit_Name);
set ex4_2;
Visitno = input(Visit_Name, visitfmt.);
run;

proc sort data=ex4_2_o;
by Visitno Subject;
run;

data ex4_2_updated;
retain Subject OE Trt_group VisitDt Visitno DIOP_OD DIOP_OS;
update ex4_2_o new_o;
by Visitno Subject;
run;

data ex4_2_updated_2(rename=(Visitno=Visit_Name));
```

```
set ex4_2_updated;
run;

proc print data=ex4_2_updated_2;
format VisitDt missDates. Trt_group DIOP_OD DIOP_OS missing. Visit_Name visitfmt.;
run;
```

Output

Obs	Subject	OE	Trt_group	VisitDt	Visit_Name	DIOP_OD	DIOP_OS
1	111	OD	Not Randomized	2011-05-19	Screening	19.6	21.5
2	112	OD	Not Randomized	2011-05-25	Screening	26.7	24.9
3	113	OS	Not Randomized	2011-06-03	Screening	17.2	18.8
4	114	OS	Not Randomized	2011-07-03	Screening	15.4	14.7
5	111	OD	Not Randomized	2011-06-29	Baseline	21.3	Not Available
6	112	OD	Not Randomized	2011-06-30	Baseline	34.6	Not Available
7	113	OS	Not Randomized	2011-07-07	Baseline	Not Available	Not Washed Out
8	114	OS	Exit	Exit	Baseline	Not Available	Exit
9	111	OD	1	2011-07-01	Postop_1 Day	18.2	Not Available
10	112	OD	2	2011-07-02	Postop_1 Day	24.3	Not Available
11	113	OS	2	2011-07-09	Postop_1 Day	Not Available	20.8
12	114	OS	Exit	Exit	Postop_1 Day	Not Available	Exit
13	111	OD	1	Lost to follow up	Postop_1 Month	Lost To Follow Up	Not Available
14	112	OD	2	2011-07-28	Postop_1 Month	21.8	Not Available
15	113	OS	2	2011-08-06	Postop_1 Month	Not Available	18.1
16	114	OS	Exit	Exit	Postop_1 Month	Not Available	Exit
17	111	OD	1	Lost to follow up	Postop_12 Month	Lost To Follow Up	Not Available
18	112	OD	2	2012-06-13	Postop_12 Month	20.5	Not Available
19	113	OS	2	2012-06-20	Postop_12 Month	Not Available	17.9
20	114	OS	Exit	Exit	Postop_12 Month	Not Available	Exit

HOW TO COUNT MISSING VALUES

Suppose that our goal is to count the number of missed DIOPs for each subject across all visits, and we have a dataset named IOP_by_v as follows (see Example 5). One way to do this is to apply the missing() function. The missing() function checks a numeric or character expression for a missing value, and returns a numeric result of either 0 or 1. Missing (variable) = 1 if there is at least one missing value with that variable; otherwise, missing(variable)=0.

Example 5: dataset IOP_by_v

Obs	Subject	DIOP_Scr	DIOP_BL	DIOP_D1	DIOP_M1	DIOP_M12
1	111	19.6	21.3	18.2	L	L
2	112	26.7	34.6	24.3	21.8	D
3	113	18.8	W	20.8	18.1	17.9
4	114	14.7	E	E	E	E

```
***Code 1 of Example 5;
data N_miss1;
set DIOP_by_v;
N_of_miss = sum(missing(DIOP_Scr), missing(DIOP_BL), missing(DIOP_D1),
missing(DIOP_M1), missing(DIOP_M12));
run;

proc print data=N_miss1;
run;
```

Output of Example 5 using Code 1

Obs	Subject	DIOP_Scr	DIOP_BL	DIOP_D1	DIOP_M1	DIOP_M12	N_of_miss
1	111	19.6	21.3	18.2	L	L	2
2	112	26.7	34.6	24.3	21.8	D	1
3	113	18.8	W	20.8	18.1	17.9	1
4	114	14.7	E	E	E	E	4

Another way to count missing values is to use nmiss() function. The following code generates exactly the same output.

```
***Code 2 of Example 5;
data N_miss2;
set DIOP_by_v;
N_of_miss = nmiss(DIOP_Scr, DIOP_BL, DIOP_D1, DIOP_M1, DIOP_M12);
run;

proc print data=N_miss2;
run;
```

We could also consider using an array (see the code 3 below) in such an example. The array can be applied multiple times to calculate different statistics (mean, min, max, etc.). It will also make our life easier if we would like to add or delete variables. In Code 3 of Example 5, a new variable DIOP_M3 (DIOP value of Postoperative 3 Month Visit) can be easily added into the array, and all the corresponding statistics will be taken care of automatically. This is similar to the way how macro benefits us.

```
***Code 3 of Example 5;
data N_miss3;
```

```
set DIOP_by_v;
array DIOP{*} DIOP_Scr DIOP_BL DIOP_D1 DIOP_M1 DIOP_M12;
N_of_miss= nmiss(of DIOP{*});
DIOP_mean= mean (of DIOP{*});
DIOP_min = min (of DIOP{*});
DIOP_max = max (of DIOP{*});
run;

proc print data=N_miss3;
run;
```

Output of Example 5 using Code 3

Obs	Subject	DIOP_Scr	DIOP_BL	DIOP_D1	DIOP_M1	DIOP_M12	N_of_miss	DIOP_mean	DIOP_min	DIOP_max
1	111	19.6	21.3	18.2	L	L	2	19.70	18.2	21.3
2	112	26.7	34.6	24.3	21.8	D	1	26.85	21.8	34.6
3	113	18.8	W	20.8	18.1	17.9	1	18.90	17.9	20.8
4	114	14.7	E	E	E	E	4	14.70	14.7	14.7

CONCLUSIONS

In clinical trials missing data could contain usefully information that we don't want to lose. Representing missing dates with proper future dates can assist date range checks. Representing missing values with special missing characters can help retain valuable information in the output. These methods help us keep as much unbiased information that is related to the missing data as we can in clinical trials before any imputation methods are applied.

REFERENCES

MISSING! - Understanding and Making the Most of Missing Data. Suzanne M. Humphreys, PRA International, Victoria, BC

How to Represent Missing Data: Special Missing Values vs. 999999999. Quentin McMullen, Westat, Rockville MD

ACKNOWLEDGMENTS

I would like to thank Peter Eberhardt for all his support and advice in editing this paper.

CONTACT INFORMATION

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

Lucheng Shao
Ivantis, Inc.
38 Discovery, Ste 150
Irvine, CA, 92618
lshao@ivantisinc.com