

Imputing Dose Levels for Adverse Events

John R Gerlach & Igor Kolodezh
CSG Inc., Raleigh, NC USA

ABSTRACT

Besides the standard reporting of adverse events in clinical trials, there is a growing interest in producing similar analyses in the context of exposure to treatment drug at the onset of an adverse event. Given an ADaM data set containing adverse events (ADAE), the intended analysis initially requires the inclusion of a variable denoting the dose level at the onset of an adverse event, called DOSEAEON. This variable would contain a null value for non-treatment emergent events, zero for placebo, and indicate the dose level, otherwise. Moreover, DOSEAEON would be used to create grouping variables for the actual analysis. This paper discusses the challenges of implementing a hierarchical methodology to determine the dose level at the onset of an adverse event.

INTRODUCTION

Reporting adverse events (AE) is a crucial component of safety analyses involving clinical trials. In fact, there are several standard reports, organized by System Organ Class (SOC) and Preferred Term (PT), across treatment groups that show the incidence of adverse events during a study. Similar reports show the severity of the adverse event and the relationship to study drug, as well. These standard reports are invaluable for the safety analysis in a clinical trial. However, except for the assigned treatment group (i.e. ARMCD) these reports do not consider the dose level at the onset of adverse events.

Consider a report that shows the incidence of adverse events in the context of dose ranges. The layout of the report emulates standard AE reports; however, the incidence represents the number of subjects having an adverse event across ranges of dose levels, rather than the usual categories, such as across treatment groups or severity (i.e. Mild, Moderate, and Severe). The following table shell below illustrates the report. Notice that the column headers denote dose ranges, such as “<50-100” and “>100-200”. The denominator used to compute percentages represents those subjects having an adverse event for a specific dose range. Thus, a subject may contribute to more than one column, since it is possible to have *different* adverse events at *different* dose levels. However, a subject would contribute *only once* for an adverse event (i.e. row), using the *highest dose level*.

Table 1.2
Subjects Having At Least 1 TEAE, By Dose Administered
Population: Phase 1 Safety Pooling

	Actual Individual Dose of Wonder Drug (mg)			
MedDRA (Version 15.0) Primary System Organ Class Preferred Term	<=50 N=XXX n (%)	<50-100 N=XXX n (%)	>100-200 N=XXX n (%)	>200 N=XXX n (%)
System Organ Class #1 Preferred Term #1	xxx (xxx.x)	xxx (xxx.x)	xxx (xxx.x)	xxx (xxx.x)
Preferred Term #2	xxx (xxx.x)	xxx (xxx.x)	xxx (xxx.x)	xxx (xxx.x)

METHODOLOGY & IMPLEMENTATION

Given two ADaM data sets, ADAE and ADEX, that contain adverse events and exposure data, respectively, the primary objective is to add one variable, DOSEAEON, denoting the dose level at AE onset, to each observation in the ADAE data set. If only treatment emergent adverse events are considered, then the target data set might be a subset of ADAE, albeit having the same variables *plus* the dose level information.

In order to impute the dose level for a given adverse event, it is necessary to consider the subject’s exposure periods. For instance, if a subject had an adverse event on 07SEP2012 at 10:31 AM, the process attempts to find an appropriate exposure period in order to assign the respective dose for that event. *But, what’s the definition of exposure period?* Is it the pair-wise start / end date-time or date values? Certainly, date-times would be used first;

otherwise, you would use dates, along with the respective AE onset date-time or date. But, what if the end date-time value is null? Even worse, what if the date-time or the dates are *the same value*? How would you discern whether the onset of an adverse event, occurring at 07SEP2012 at 10:31 AM, falls within an exposure period consisting of a *single time point*, such as 07SEP2012 at 10:30 AM? You can't. Thus, the methodology would consider the *next* start date-time or date, as part of the exposure period. Obviously, preliminary investigation about the longitudinal data points becomes necessary in order to formulate a reasonable methodology.

The following code creates a report that shows how longitudinal data are populated with respect to each other; that is, whether the variables contain null or actual values.

```
proc format;
  value nullf . = 'Null'
             other = 'Not Null';
run;

proc freq data=ex01;
  tables exsdtm * exedtm * exsdt * exedt / list missing;
  format _numeric_ nullf.;
run;
```

Reasonably, the proposed methodology is hierarchical. First, it is necessary to know whether the event is viable, occurring after a subject's first dose, that is, a treatment emergent event. Assuming a viable event, the process attempts to find an appropriate exposure period; otherwise, the last dose might be used. Certainly, the methodology must be clinically sound. However, without proper preliminary analysis of the longitudinal data points, a proposed methodology can become extremely convoluted, fundamentally flawed, and perhaps impossible to implement.

In regards to implementation, initially, two disparate data sets (ADAE and ADEX) must be combined hoping to facilitate the task. One method might employ the SQL procedure with the BETWEEN / AND operators. However, considering the hierarchical nature of the methodology, along with SQL's Cartesian product, the joins would likely compound the task. Another method might transpose the AE data and use several Data steps focusing on different components of the hierarchical methodology, a kind of divide and conquer approach. Obviously, the implementation can make the task almost untenable, especially if the requirements are modified later, per request from clinicians and biostatisticians.

The proposed solution takes a more natural approach by recognizing the fundamental objective: initially to add one variable to the adverse events data set; that is, to assign a dose level, denoted by the variable DOSEAEON, to each adverse event. Thus, the target data set contains the adverse events, plus one primary variable, DOSEAEON, and its respective dose grouping variables (e.g. DOSEGR1N). Therefore, it seems more natural to transpose the exposure data and place it in juxtaposition with the adverse events in order to create the target data set called ADAE_DOSE. More importantly, this method allows the use of parallel arrays that represent Start / End date-times and dates, treatment, and dose, which facilitate the intricate process of imputing the dose level at AE onset.

The proposed method of juxtaposing exposure and adverse events initially requires that both ADaM data sets are sorted longitudinally, especially the exposure data. The PROC steps, shown below, accomplish the task, as well as keeping only the variables needed to accomplish the task. Moreover, selection criteria may be implemented at this point, using a WHERE statement, such as targeting a specific study population, including treatment emergent adverse events, or excluding certain studies in a large ISS study. Ultimately, the data set representing adverse events will determine the cardinality of the target data set; whereas, the exposure data is supplemental.

```
proc sort data=ad.adae(keep=studyid usubjid aeseq trtemfl aesdt aesdtm) out=ae01;
  by studyid usubjid aeseq;
run;

proc sort data=ad.adex(keep=studyid usubjid exseq exdose exsdt exedt exsdtm exedtm)
  out=ex01 nodupkey;
  by studyid usubjid exsdt exsdtm exdose;
run;
```

EVENTS OCCURRING BEFORE AND AFTER EXPOSURE

Assigning a dose level at AE onset implies that the event took place during an exposure period. However, the hierarchical methodology first asks whether the adverse event is viable; that is, whether it occurred after the first dose, by comparing the AE onset with the first dose using date-time or date information, for example, the AE start date-time and the date-time of the first dose. The first dose information can be stored in two formats, FDTIME and FDATE that map a subject identifier to the first dose date-time and first dose date, respectively. Several Data steps create Control Input data sets that contain the needed information, as shown below.

```

data fmts;
  merge ae(in=ae keep=studyid usubjid) ex(keep=studyid usubjid exsdt exdtm exdose);
  by studyid usubjid;
  if ae;
run;

data fdate(rename=(usubjid=start exsdt=label));
  retain fmtname 'FDATE' type 'I';
  set fmts(keep=usubjid exsdt);
  by usubjid;
  if first.usubjid
  then output;
run;

data fdtime(rename=(usubjid=start exsdtm=label));
  retain fmtname 'FDTIME' type 'I';
  set fmts(keep=usubjid exsdt);
  if first.usubjid
  then output;
run;

proc format cntlin=fdate; run;
proc format cntlin=fdtime; run;

```

The following code segment shows how the formats would determine the viability of an adverse event. If the AE onset date-time or date occurs after the first dose, then the process proceeds to determine the dose level at AE onset by traversing the exposure periods.

```

if (aesdtm ne. and aesdtm lt input(usubjid,fdtime.))
or (aesdt ne . and aesdt lt input(usubjid,fdate.))
then < Not Viable >
else < Determine Dose Level >

```

Two other formats LDTIME and LDATE are created in a similar fashion to address the possibility that an adverse event occurred *after* the last dose, which will be discussed later. In this situation, it is necessary to know the value of the last dose, as well; hence, the additional format LDOSE.

```

* LDATE -- Similar DATA & PROC steps used for creating FDATE format ;
* LDTIME -- Similar DATA & PROC steps used for creating FDTIME format ;

data ldose(rename=(usubjid=start exdose=label));
  retain fmtname 'LDOSE' type 'I';
  set fmts(keep=usubjid exdose);
run;

proc format cntlin=ldose;
run;

```

JUXTAPOSING ADVERSE EVENTS AND EXPOSURE

Initially, the task of juxtaposing the exposure data to the adverse events requires a transposition of the exposure data set in order to define the vectors, specifically: Start / End date-times and dates, treatments, and dose. However, in order to define the dimension of the arrays, it is necessary to determine the maximum number of exposure records across subjects. The FREQ procedure below obtains the number of exposure records per subject, then the SQL procedure creates a macro variable needed to define the parallel arrays (e.g. SDT1-SDT&nex.).

```
proc freq data=ex01;
  tables usubjid / noprint out=exmax;
run;

proc sql noprint;
  select max(count) into :nex
  from exmax;
quit;
%let nex = &nex.;
```

A single Data step achieves the goal of putting all the data points in a single observation, using an intuitive naming convention for the parallel arrays. Thus, for example, if there were at most fifty exposure records, then the required array variables include: SDTM1-SDTM50; EDTM1-EDTM50; SDT1-SDT50; EDT1-EDT50; TRT1-TRT50; DOSE1-DOSE50.

```
data ex02;
  length trt1-trt&nex. $20;
  retain sdt1-sdt&nex. edt1-edt&nex. sdtm1-sdtm&nex. edtm1-edtm&nex.
  trt1-trt&nex. dose1-dose&nex.;
  array sdt{*} sdt1-sdt&nex.;
  array edt{*} edt1-edt&nex.;
  array sdtm{*} sdtm1-sdtm&nex.;
  array edtm{*} edtm1-edtm&nex.;
  array trt{*} trt1-trt&nex.;
  array dose{*} dose1-dose&nex.;
  set ex01;
  by usubjid;
  if first.usubjid
  then do;
    item = 1;
    do i = 1 to &nex.;
      sdt{i} = .;      edt{i} = .;
      sdtm{i} = .;    edtm{i} = .;
      trt{i} = '';    dose{i} = .;
    end;
  end;
  else do;
    sdt{item} = exsdt;  edt{item} = exedt;
    sdtm{item} = exsdtm; edtm{item} = exedtm;
    trt{item} = extrt;  dose{item} = exdose;
    item + 1;
  end;
  if last.usubjid
  then output;
  keep studyid usubjid sdt1-sdt&nex. edt1-edt&nex. sdtm1-sdtm&nex.
  edtm1-edtm&nex. trt1-trt&nex. dose1-dose&nex.;
run;
```

One might consider using the TRANSPOSE procedure. However, there's the issue of transposing numeric and character data, thereby ending-up with all character (COLn) variables. Even worse, the date-time and date values would be internal SAS values. Worst of all, the transposed data set does not have the desired structure; that is, having all the data points in a single observation.

CREATING THE TARGET DATASET

A simple Match-Merge combines the adverse events with the transposed exposure data. The sub-setting IF statement ensures the inclusion of only those subjects having at least one adverse event, since it is conceivable for a subject to have taken the study drug, but not have an adverse event. The FORMAT statement assigns Date and Date-Time formats to the variables, respectively, for development purposes only. Ultimately, the data set AEEX contains the same number of observations as the data set AE01, as shown in the Data step below. However, there are many more variables, specifically, *sets of variables* denoting: Start / End Date-Time; Start / End Date; Treatment; and Dose Level. Moreover, for each set, the number of variables represents the highest number of exposure records. Consequently, it is a large data set; however, extremely manageable and well-suited for longitudinal processing using parallel arrays.

```
data aeex;
  merge ae01(in=ae) ex02;
    by studyid usubjid;
  if ae;
  format sdt1-sdt&nex. edt1-edt&nex. date9. sdtm1-sdtm&nex. edtm1-edtm&nex. datetime15.;
run;
```

The process proceeds to traverse the exposure periods in an attempt to find the dose level at the onset of each adverse event. The following Data step defines parallel arrays representing the longitudinal data, treatment, and dose level. In accordance with the hierarchical methodology, the Data set determines the viability of the adverse using the formats FDATE and FDTIME. Assuming that the event occurred sometime after first dose, the DO WHILE loops seeks to find an appropriate exposure using nested IF statements, employing first date-time elements, if available, otherwise date elements. Once the nested logic determines that an adverse event occurred during a specific exposure period, further logic determines the dose level based on the treatment at that time.

```
data dosevars;
  set aeex01;
  array sdt{*} sdt1-sdt&nex.;
  array edt{*} edt1-edt&nex.;
  array sdtm{*} sdtm1-sdtm&nex.;
  array edtm{*} edtm1-edtm&nex.;
  array trt{*} trt1-trt&nex.;
  array dose{*} dose1-dose&nex.;

  doseaeon = .;

  if (aesdt ne . and aesdtm lt input(usubjid,fdate.)
    and (aesdtm ne . and aesdtm lt input(usubjid,ftime.)
    then < Not a viable Adverse Event >;

  else do;
    i = 1;
    do while(doseaeon eq . and i le (&nex.-1));

      < Obtain Dose Level Using Date-Time Elements
        Otherwise, Use Date Elements >

      if doseaeon ne .
        then leave;
      i+1;
    end;
```

Typically, the variable DOSEAEON contains three possible values: zero denoting Placebo; an actual dose value; or, a null value indicating that the event is not considered for analysis. If the event does not occur during any exposure period, then the aforementioned formats LDATE and LDTIME determine whether the adverse event occurred after the last dose; whereupon, an INPUT function uses the LDOSE format to obtain the dose level.

```

if doseaeon eq .
  and ( aesdt ge input(usubjid,ldate.) or
        aesdtm ge input(usubjid,ldtime.) )
  then do;
    if input(usubjid,dose.) eq 0
      then doseaeon = 0;
    else doseaeon = input(usubjid,ldose.);
  end;
end;
if doseaeon ne .
  then < Assign Dose Grouping Variable(s) >
keep studyid usubjid aeseq doseaeon < Dose Grouping Variable(s) >;
run;

```

Finally, the Data step proceeds to create Dose Grouping variables that are used for the actual analysis. For example, the variable DOSEGR1N is assigned values using the Informat DOSE1LVL that maps ranges of dose values to a numeric categorical value. Then, the variable DOSEGR1 is assigned values using the Format DOSE1LVL that maps the categorical value to the text range value. Notice that both formats have the same name.

```

proc format;
  invalue dos1lvl 0 = 0
                 2.5 - 50 = 1
                 50 <- 100 = 2
                 100 <- 200 = 3
                 200 <- 400 = 4
                 other = 5;

  value dos1lvl 0 = 'Placebo'
               1 = '<= 50'
               2 = '>50 to <=100'
               3 = '>100 to <=200'
               4 = '>200 to <=400'
               5 = 'Over 400';
run;

```

Keep in mind that the data set DOSEVARS contains only primary key variables and the dosing information. The following Data step creates the target ADaM data set by merging the DOSEVARS data set with the original AD.ADAE data set.

```

data ad.adae_dos;
  merge ad.adae dosevars;
  by studyid usubjid aeseq;
  label doseaeon = 'Trt Dose Strength at Time of AE (mg)'
        dosegr1n = 'Dose Group 1 (N)'
        dosegr1 = 'Dose Group 1';
run;

```

VALIDATION

Like most tasks involving data analysis of clinical trials, there is a validation component to the overall process. Given the Target and Validation data sets, the COMPARE procedure easily accomplishes the task to ensure integrity of the deliverable. Of course, unless specified otherwise, the COMPARE procedure compares *every* variable in both data sets. Thus, initially, the validation process should focus just on the DOSEAEON variable. Even the respective categorical variables might be considered superfluous, since if DOSEAEON does not match, then certainly the respective categorical variables (e.g. DOSEGR1N) will likely not match, thereby generating unnecessary output. Consider the COMPARE procedure below that includes only those variables that identify the record (USUBJID and AESEQ) and the primary target variable (DOSEAEON).

```

proc compare base      = ad.adae_dos(keep=usubjid aeseq doseaeon)
              compare = qc.adae_dos(keep=usubjid aeseq doseaeon);
  id usubjid aeseq;
run;

```

Of course, eventually, both data sets should be completely evaluated. Although DOSEAEON may be correct, it does not preclude the possibility that there's an issue with the respective categorical variables (e.g. DOSEGR1N), or even the tag-along variables from the ADAE ADaM data set.

Validation issues are often systemic usually caused by a flaw in the implementation of a rule. Unfortunately, the COMPARE procedure generates detailed information that does not indicate the systemic problem. For example, it is not helpful knowing that the BASE, COMPARE, and DIFF values are: 100, 0, and -100, respectively, for a particular subject. Rather than pore over instances of discrepancies *ad nauseam*, consider the following summary report that locates the discrepancy in the context of Dose Levels.

DOSEAEON	TARGET	VAL	DIFF
.	320	510	-190
0.0	200	180	20
2.5	30	0	30
10.0	40	40	0
20.0	50	0	50
25.0	50	50	0
50.0	190	160	30
75.0	115	115	0
80.0	40	40	0
100.0	130	130	0
150.0	160	160	0
200.0	760	700	60
300.0	20	20	0
400.0	140	140	0
	=====	=====	=====
	1545	1545	0

Basically, the report shows the number of events distributed across the actual Dose Levels, as found in the Target and Validation data sets. The final column DIFF indicates the difference (TARGET minus VAL). Obviously, the totals of the TARGET and VAL columns should match, indicating the number of adverse events per dose level. The third column represents the algebraic sum whose total is always zero. Notice the null value of DOSEAEON, for which there is a substantial discrepancy; whereas, the other discrepancies are subsets. In other words, there seems to be an issue when imputing a null value for DOSEAEON. The code for producing this report requires two FREQ steps generating two data sets, a Data step followed by the PRINT procedure.

```

proc freq data=dosevars;
  tables doseaeon / noprint out=qc(keep=doseaeon count rename=count=qc);
run;

proc freq data=ad.adae_dose;
  tables doseaeon / noprint out=target(keep=doseaeon count rename=count=target);
run;

data rep;
  merge target qc;
  by doseaeon;
  if target eq . then target = 0;
  if qc      eq . then qc      = 0;
run;

proc print data=rep;
  id doseaeon;
  sum target qc diff;
run;

```

CONCLUSION

Reporting the incidence of adverse events in the context of dose levels gives further insight into the effects of a study drug, especially with respect to titration studies. However, determining the dose level at the onset of an adverse event can be quite a challenge depending on the methodology and implementation, especially when dealing with incomplete start / end data points or unexpected changes in the analysis. Crossover studies offer a greater challenge since an adverse event can occur during a treatment that is not the study drug. In this case, the rules must specify the time window in order to determine whether the event may be attributed to the study drug, even though it may be in closer proximity to another treatment. Whatever the situation, the proposed hierarchical methodology processes each adverse event by considering the collection of exposure data germane to a subject, where date-time has priority over dates. Assuming a reasonable methodology and a robust implementation, the target data set contains the content of the ADaM ADAE data set, along with the dosing variables denoting the actual dose level at onset and respective variables denoting dose groups.

REFERENCES

Cao, Dongsun & He, Xiaomin; "Statistical Analysis of Adverse Events in Randomized Clinical Trials Using SAS." *Proceedings of the Pharmaceutical SAS User's Group Conference, 2011.*

CDISC ADaM Data Structure for Adverse Event Analysis Version 1.0
<http://www.cdisc.org/adam>

Lu, Zaizai and Shen, David; "Dynamic Data Matching in Clinical Trial Research." *Proceedings of the SAS User's Group International Conference, 2002.*

ACKNOWLEDGMENTS

Special thanks to Phyllis Smetana of UCB Pharmaceuticals for her participation in the development of this paper.

CONTACT INFORMATION

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

Name:	John R. Gerlach & Igor Kolodezh
Enterprise:	CSG, Inc.
Address:	7780 Brier Creek Parkway
City, State ZIP:	Raleigh NC 27617
Web:	www.csg-inc.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.