PhUSE 2010

Paper DH02

How to clean up dirty data in Patient reported outcomes

Knut Mueller, UCB – Schwarz Biosciences, Monheim, Germany

ABSTRACT

The current FDA “Guidance for Industry - Patient Reported Outcome Measures” from December 2009 describes the assessment of patient reported outcome variables in clinical trials. One important rule is that information coming from the patient need to be recorded in the same way as they were answered by the patient. The investigator is not allowed to query any of the patient's answers which leads in general to a lot of dirty data. Based on the rules mentioned in the guideline there are also several restrictions for the data workflow starting from data entry, to data management and statistical reporting. This paper briefly describes a proposal for a guideline compliant workflow and gives a proposal for an algorithm how to handle ambiguous or unclean patient reported outcome data with a validated SAS macro according to pre-specified data entry and counting rules

INTRODUCTION

Patient reported outcomes (PROs) are typically assessed using questionnaires. According to the aforementioned FDA guidance (1), the completed questionnaires are considered to be source documents comparable to case report forms (CRFs). The main difference is that once the patient has left the site, it is not possible to query data that is not exactly according to the completion rules. The information given by the patient on the questionnaire must not be changed. Nevertheless, all questionnaire entries have to be considered as the patient's response. Since patients do not necessarily always follow the entry rules this leads to a lot of "dirty" data.

In the following a proposal for a guideline compliant workflow will be described and an algorithm how to handle ambiguous or unclean patient reported outcome data is presented. This algorithm is later to be integrated in a validated standard macro. Standard macros can facilitate the production of SDTM and analysis datasets when complex algorithms have to be repeatedly applied across studies in a similar manner. However, the maintenance of standard macros requires a considerable amount of resources. Thus such a macro should be circumscribed closely enough in order to remain both easy to use and easy to maintain.

SITUATION

The decision was made that all information provided by the patient while filling in a questionnaire will go into the database without applying any rules to clean up that data beforehand. It was decided that the application of the counting rules will be done by Statistical programming during the creation of SDTM and ADaM datasets. It was decided not to do it at the level of data entry as the rules are quite complex and it would result in a loss of traceability.

As a result the database contains entries that are not according to the entry rules like having additional comments, out of range answers or multiple answers to items where only one single answer is allowed. Nevertheless all these data are still informative but need to be cleaned before they can be used for statistical analysis. This requires specific rules for storing this unclean data in the database. Especially the way multiple answers are stored has to be clearly defined.

Multiple answers are the most common type of unclean PRO data. They will be entered as numbers divided by a slash into the respective character variable (e.g. "2/3", "1/5", "2/3/4").

The rules for handling unclean data are set up by the Health Outcomes department, who are the specialists for PRO interpretation and analysis, in collaboration with Biostatistics.

The following counting rules are usually described in the statistical analysis plan for a study:

Specific counting rules

The following rules will apply for analysis of (1) out of range and (2) ambiguous answers (i.e. invalid or unable to interpret answers) to questionnaires completed by subjects:

1. In case of an out of range answer (i.e., an answer that does not correspond to any possible response proposed in the questionnaire, e. g. “?”, “I don’t know” or any value superior or inferior to the ones specified in the

response options): the answer will be scored “missing”.
However, in case the subject selected one of the proposed responses but added a comment (for instance “6 +++” or “5 ?”), the response (i.e., "6" or "5") will be retained for scoring but not the comment (i.e., "+++" or "?").
In the same way, if the subject selected one of the proposed responses but added a value superior or inferior to the ones specified in the responses options (for instance “4/5” or “-1/2” on a 5-point scale ranging from 0 to 4), the response corresponding to the possible responses options (i.e., "4" or "2") will be retained for scoring but not the values superior or inferior to the responses options (i.e., "5" or "-1").

2. In case of an ambiguous answer (i.e., multiple responses to a question allowing only a single response, a response marked between two allowed responses):

2.1. Multiple responses to a question allowing only a single response:
If more than half of the responses are marked (i.e., 4 responses marked on a seven point scale, 3 responses marked on a 5-point scale, 3 responses marked on a 4-point scale, 2 responses to a Yes/No item…): the answer will be scored “missing”.
If half or less than half of the responses are marked:
if the responses are NOT adjacent to each other: the answer will be scored “missing”,
if the responses are adjacent to each other (“2/3” or “2/3/4”, for instance), the more severe score will be retained.

2.2. If a response is marked between two allowed responses (for instance, the subject marked his/her response between 2 and 3 on a 4-point scale allowing only responses 1, 2, 3 and 4), the same rule applies as if two adjacent answers were marked.

These general rules apply to all PROs that are once identified. However, they have nevertheless to be explicitely be specified in the Statistical Analysis Plan for every PRO within every study where they are to be applied.
Similarly, the standardized questionnaires that qualify for this rule will be identified by the Health Outcomes department in collaboration with the Biostatistics department. In general all PROs with a response format where the numeric responses are ordered by severity and the steps are equidistant could be handled.
It is not applicable to visual analog scales and to scales where the most severe value cannot be identified or the most severe value is neither the minimum nor the maximum of the numeric response.

A questionnaire that qualifies for this rules is the SF36 Health Survey. An example for an item of the SF 36 is item MH02 (item 9c) of the SF 36 Health Survey: "Have you felt so down in the dumps that nothing could cheer you up?" with the response categories: 1 = All of the time, 2= Most of the time, 3 = Some of the time, 4 = A little of the time, 5 = none of the time. The application of the rules mentioned above to this item means that a maximum of two adjacent responses will be replaced by the lower response. If there is a response with three answers or with two answers that are not adjacent, it will be set to missing.

APPROACH

ASSUMPTIONS

All the information coming from the patient has to be stored in the database. In order to deal with PRO data that might contain ambiguous answers, the data entry rules have to be specified carefully. For example multiple answers have to be entered as numbers divided by a slash in order to be able to identify them and to handle them according to the pre-specified rules.

REQUIREMENTS FOR STATISTICAL PROGRAMMING

To clean the raw data for analysis according to the counting rules, a standard macro will be provided.
In daily work, not all statistical programmers like using standard macros. Standard macros are often not accepted because they are too complicated, in terms of what they do and what requirements they have. Especially when macros call lots of other macros, etc. The loss of control over the process causes often discomfort in using complex standard macros. Therefore one of the most important criteria for the development of a standard tool to cope with unclean PRO data was that is user friendly. That means the amount of parameters should be limited to a small number and they should be self explaining and well documented. The scope, as well, should be limited, the result of what it did should be traceable, documentation should clearly state what it does, it should of course restore the environment as it was before. And due to a limited complexity re-validation after necessary updates should be done within reasonable time and effort.

SCOPE

In order to meet the above mentioned criteria we decided to create a macro that checks item by item of a questionnaire where the above mentioned general rule is applicable. It first reads the original value, then applies the specified rules where appropriate and at a last step imputes either a valid value or a missing value. Already valid values remain untouched. Due to tracing reasons a comment is written into an additional variable. This variable can identify the cases that have been manipulated and have to be checked manually. Additionally a flag is set to identify

cases for a sensitivity analysis. The original data will be kept to ensure traceability.
This macro has to be called for each item. It does not check the whole questionnaire all at once. That might lead to a considerable amount of macro calls, in the case of he SF36 the macro has to be called for all of the 36 items. On the other hand the amount of macro parameters that have to be specified is kept low, and the user still has some control over the process. What the macro does is limited to one specific task that is imputing ambiguous answers inside one specified item. The source data has to be normalized, thus this tool can best be applied within SDTM or ADaM development. Its concept is similar to a SAS call routine with an integrated "WHERE" statement. Due to its call-routine-like character the macro cannot stand-alone. It has to be called in a datastep.

PROGRAMMING TECHNIQUES

The described rogramming techniques will focus on the use of Perl regular expressions (PRX) to identify how a certain entry has to be treated (2). Of course, some of the tasks could as well be done with other SAS functionalities, however, PRX are the most efficient way to manipulate text strings (3) . The presented code examples are small samples to illustrate the algorithms used to identify how ambiguous or unexpected data has to be treated. They are not necessarily comprehensive.

PROCESS DESCRIPTION

This section describes what steps are necessary to deal with unclean data. It does not describe the macro as a whole and does not describe the steps taken to ensure a proper functioning across study environments etc. The steps describe how the data entries are checked and what decisions are taken within each step (see figure 1).

MACRO PARAMETERS

The macro variables that have to be specified in the macro call are:

TEST	the parameter that contains the name of the variable that contains the item names
ITEM	the parameter that contains the name of the item
INVAR	the parameter that contains the input variable. The input variable has to be character format and contains the patient's answer to the respective item as a numeric representation, including multiple answers or comments or out of range variables.
VALUES	the parameter that contains the list of valid values for that particular item separated by a blank.
SEV	the parameter that specifies if the most severe value is the highest or the lowest in this list.
OUTVAR	the parameter that contains the name of the output variable
COMMENT	the parameter that contains the name of the variable where the comment is written to.

DERIVED MACRO VARIABLES

From VALUE the maximum number of multiple answers will be derived. The macro itself derives a bunch of other variables needed to specify the regular expressions used to scan the input variable. The most important being

NAMBI	the maximum of multiple answers for this item.
REPEAT	the maximum number of multiple answers for this item minus one
REGXP	a Perl regular expression describing valid answers.

NREGXP a Perl regular expression describing out of range answers.

INTERIM VARIABLES

A number of auxiliary variable are created that are used during the imputation and that will be removed at the end. The most important to mention are _INVAR which contains the value of the original &INVAR, and _OUTVAR which contains the final result before it is written to &OUTVAR.

MACRO CALL

A macro call, for example, for item MH02 of the SF36 Health ("Have you felt so down in the dumps that nothing could cheer you up?" with the response categories: 1 = All of the time, 2= Most of the time, 3 = Some of the time, 4 = A little of the time, 5 = none of the time) could look like:

```
%proimpute
(test=QSTESTCD,item=MH02,invar=qsorres,values= 1 2 3 4 5,sev=low,outvar=qsstresn);
```

In this example for item MH02 of the SF 36 the macro variable NAMBI resolves to 2, REPEAT resolves to 1, REGEXP resolves to [123455] and NREGEXP resolves to (-\d\d?|[06789]|\d\d+). The variable _INVAR is set to the values of qsorres.

STEP1

First answers that already are missing or contain a code representing a missing value ("Not done", "ND" "unknown", "UN", "not applicable", "NA") are identified, and excluded from the next steps.

STEP2

In a second step, correct values are identified, which remain untouched. The first two steps can be done using standard data step statements.

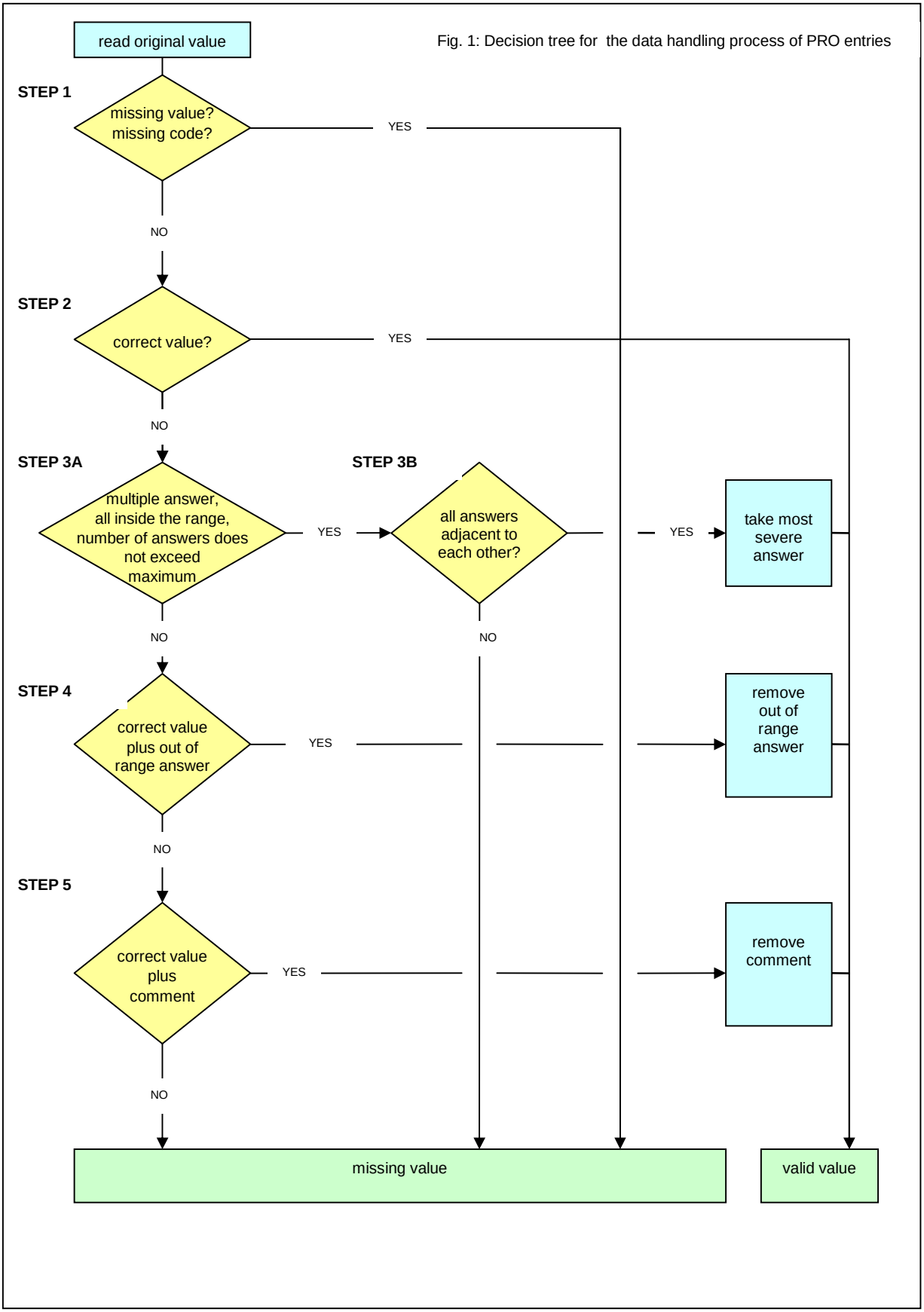

Fig. 1: Decision tree for the data handling process of PRO entries

STEP 3A

In the next step, we identify multiple answers where all answers are inside the range and the number of answers does not exceed the allowed maximum number of answers (i.e no more than half the number of possible answer categories). In our example that are responses like "1/2", "3/5", 3/4".

```
if prxmatch("/^(&regxp.)(\/(&regxp.)){0,&repeat.}$/",strip(&invar.)) then do;
```

Answers with more than two responses will be ruled out and set to missing ("2/3/4", "1/2/5")

STEP 3B

Then, it will be checked if the answers are adjacent to each other. If so the most severe will be taken. In our example the answers "1/2" and "3/4" are identified as adjacent and will be replaced by the most severe answer ("1" and "3" respectively). If they are not adjacent the result will be set to missing.

```
%DO k=1 %TO &nambi.;
   answer&k=input(scan(strip(&invar.),&k.,'/'), 8.);
   %IF &k = 1 %THEN %DO;
       _outvar1 =answer1;
   %END;
   %IF &k ne 1 %THEN %DO;
      %LET l =%EVAL (&k-1);
      if answer&k ne . then do;
         if abs(answer&k - answer&l) =1 and adj ne 0 then adj=1;
         else adj=0;
      end;
      else adj=adj;
      if adj=1 then do;
         if sev="high" then _outvar = max(_outvar1 ,answer&k);
         else if sev="low" then _outvar = min(_outvar1 ,answer&k);
         comment="adjacent multiple answers are replaced by the most severe value";
      end;
      else if adj=0 then do;
         _outvar=.;
         comment="multiple answers, not adjacent to each other are set to missing";
      end;
   %END;
%END;
```

STEP 4

The next step identifies multiple answers that contain a correct value and an out of range value. The out of range values will be removed and the correct value will be kept. For example "3/7" will resolve to "3"

```
if _n_= 1 then prxid1=prxparse("/\/(&REGXP.)/");
retain prxid;
/*IDENTIFY outside range value before the string*/
if prxmatch("/^(&nregxp.)\/(&REGXP.)$/",strip(_invar)) then do;
   call prxsubstr(prxid1,strip(_invar),_pos,_lgth);
   _pos=_pos+1;
   _lgth=_lgth-1;
   _outvar=input(substr(strip(_invar),_pos,_lgth),8.);
end;
/*IDENTIFY outside range value after the string*/
else if prxmatch("/^(&REGXP.)\/(&nregxp.)$/",strip(_invar)) then do;
   call prxsubstr(prxid1,strip(_invar),_pos,_lgth);
   _lgth=_lgth-1;
   _outvar=input(substr(strip(_invar),_pos,_lgth),8.);
end;
```

STEP 5

The fifth step is to identify answers with a correct value with an additional comment (for example "+/-5", "6++"). The comment will be removed and the correct value will be taken.

```
/*IDENTIFY VARIABLES WITH COMMENTS ADDED TO A NUMBER*/
```

```
else if prxmatch("/^(\D*[^-]+)*(&regxp.)\D*$/",strip(_invar)) then do;
   call prxsubstr(prxid1,strip(_invar),_pos,_lgth);
   _outvar=input(substr(_invar,_pos,_lgth),8.);
end;
else do;
   put "multiple answer with more than half of the possible answers";
   _outvar=.;
end;
```

CAVEATS

There will be still cases with unexpected values, mostly cases with a combination of comments and multiple answers which cannot be handled in a standardized way. These still have to be taken care of in a case by case basis. However these cases are identified and a comment is written to the comment variable. Thus they can be discussed during the data review meeting, for example.

CONCLUSION

- The standard approach identifies and classifies different categories of ambiguous data and imputes valid values into the results variable by applying the prespecified rules. However, the imputed values still have to be checked manually.
- The described workflow is compliant to the FDA Guidance for the use of PROs in the drug development process
- Once validated, the macro will save considerable time as the handling of dirty PRO data is one of the most time consuming parts in creation of SDTM and ADaM datasets.
- Still this approach provides a considerable amount of control over the process of generating output out of the patient's input
- Processes inside the macro have to be specified and documented very carefully.

REFERENCES

(1) U.S. Department of Health and Human Services - Food and Drug Administration (2009): Guidance for Industry - Patient-Reported Outcome Measures: Use in Medical Product Development to Support Labeling Claims

(2) Mueller K (2009): The use of Perl regular expressions (PRX) in SAS to search and modify text strings. PhUSE SDE Brussels April 2nd 2009

(3) Cody R (2004) An Introduction to Perl Regular Expressions in SAS 9, SUGI29 Paper 265-29

CONTACT INFORMATION

Knut Mueller
UCB
Alfred-Nobel Str 10
D-40789 Monheim / Germany
Work Phone: +49 (0) 2173-482054
Fax: +49(0)2173-481947
Email:knut.mueller@ucb.com
Web:www.ucb.com