Successful Lab Result Conversion for LAB Analysis Data with Minimum Effort

Pushpa Saranadasa , Merck & Co., Inc.

INTRODUCTION

In the pharmaceutical industry, the statistical results of a clinical trial's safety analysis can provide key indicators related to the safety and approval of the drug. In addition, in some of the therapeutic areas the laboratory data plays a significant role in determining the safety and/or efficacy of the drug. In a typical clinical trial, domestic or international, often a problem occurs in which the collected lab data values are reported in different units based on the country or site where the lab tests were completed. Even within the same site, different lab units, depending on whether the results were reported by central or supplemental (or local) labs, takes place. For analysis and reporting, it is now standard practice to produce safety analysis tables and reports using one of three lab standard units such as conventional, original, and/or standard international units.

This paper explores the handling of different values of units in the LAB Domain (LB) ; their conversion to international and conventional units; and effectively combining them back into one consolidated dataset. At the end of the process the original dataset remains intact, while a dataset used for analysis contains the results of the conversion process.

On the surface the presentation here seems fairly narrow, but there are some interrelated issues. Some examples come to mind immediately. Across countries the mandated units of measurement may differ, so the analysts may need to change units of measure depending on for whom the analysis is being conducted. There are two calibration and reporting issues: First, equipment and personnel used to take measurements may convert numerical results to a categorical result in which the specific meaning of, say, 'high', 'medium' and 'low' is not clear. Once there is agreement on how high, medium and low are to be normalized then conversion to statistically usable data can proceed. Secondly, across data trial sites the data may be collected by different persons using different equipment. The same person using the same blood pressure cuff may get different readings for the same patient within minutes of the different measurements. Or, the same patient visiting two clinical sites may display different readings because the different cuffs were not calibrated to one another. Implicitly, all of these issues could be dealt with in the context of this paper, however going into each one of them in detail would make the paper overly long so they are not dealt with explicitly.

PRELIMINARIES, SET-UP AND ILLUSTRATION

The uniform presentation of these different standards across collected units requires a rescaling to standard units. In this paper, data from CDISC standards and controlled terminology are used which includes lab data, although the process of conversion illustrated here is not confined to use with CDISC compliant data. The vertical structure of the LB domain of a CDISC dataset makes it straightforward to achieve the goal of rescaling the original data to the two alternative forms, international and conventional, while at the same time preserving the original data.

The variables used in the illustration for the conversion are: LBSPEC, LBORRES, LBORRESU, LBORNRHI and LBORNRLO; they are the column headers in Table 1. As per the

standards, they are in character format which creates a problem for statistical analyses. Sometimes the character format may go beyond the character representation of a real number. The fourth row of Table 1 shows an example of this. For a variety of reasons a result may be presented as an interval, as in "<8" in the table. It is important to note in this regard that CDISC does not address the question of standard units when the data is categorical as in this example, making the conversion process presented here even more important. Additionally, the lab dataset has a vertical structure as shown below in Table 1.

Looking at the data displayed in Table 1, they were collected in different units as indicated in the column labeled LBORRESU.

Table 1: Example of a Lab Data Set with Different Collected Units

LBSPEC	LBTESTCD	LBORRES	LBORRESU	LBORNRLO	LBORNRHI
plasma	AST	16	IU/L	8	22
plasma	AST	1.5	IU/dL	.8	2.2
plasma	AST	< 8	mIU/mL	8	22
plasma	AST	.4008	microkat/L	.1336	.3664
plasma	CHOL	112	mg/dL	125	200
plasma	CHOL	6.216	mmol/L	3.2375	5.18
plasma	CHOL	3.4965	mmol/l	3.2375	5.18

In most cases, the collected and desired units are expected to be in the same unit of measure. In this instance, "desired" refers to the way in which lab units are normally presented. In other words, the reference here is to the conventional way of representing the data.

For example, the unit of conventional measure for AST (rows two through four of Table 1) is IU/L, while for CHOL (rows five through seven), the conventional unit of measure is "mg/dL". Yet the table shows that the reported and conventional units are not always the same.

If the collected and desired units are the same, for example if the AST test is collected in "IU/L " unit and CHOL is collected in "mg/dL," the conversion process is much easier. In such a case, only one conversion that is from desired units to standard units may be accomplished Due to the differences in collected units, often two conversions are required in order to get the desired and standard units. Performing the conversion should have no impact on collected units as reported in the original data.

STEPS TO BE TAKEN BEFORE THE CONVERSION

Most safety analysis includes Central and Supplemental lab results. Specifically, the predefined limits of changes to the results must be retained, and every study should report these changes for each patient because they include how some lab values changed over the study period. Some studies use only central lab results for efficacy, but this may vary from study to study. Therefore, a clear understanding of which lab tests will be included in the final analysis dataset is necessary and can be achieved by early communication of this information to the study statistician and clinical group while

defining the table package and the dataset specifications. This will eventually save the time of the programmer as well as the statistician/clinical group working with the programmer on the study.

Step 1 - Get a simple frequency output

```
Proc freq data=lab noprint;
tables lbspec*lbtest* lborresu/out=units
run;
```

Using **PROC FREQ** shows how the data was collected and if there were different lab units within a lab test. It is recommended to check the units with upper and lower limits to determine whether there is a data entry issue or whether the collected data uses different units.

Step 2 - Output the dataset to a spreadsheet

Step 3 - Rename LBORRESU as "original unit"

Step 4 - Assign names for four empty columns in the spreadsheet as: conventional unit, standard unit, cfactor and sfactor

Conventional units have been used for collected results and can be rescaled to standard units as shown below in Table 2 along with their conversion factors; contacting the clinical group to request a finalized spreadsheet is highly recommended as it will be used as a source to convert results.

For purposes of illustration, the spreadsheet might appear as in Table 2 below.

.

Table 2: Example of a spreadsheet with units and their conversion factors

LBSPEC	LBTEST	original unit	conventional unit	standard unit	cfactor	sfactor
plasma	Aspartate Aminotransferase	IU/L	IU/L	microkat/L	1	0.0167
plasma	Aspartate Aminotransferase	IU/dL	IU/L	microkat/L	10	0.1667
plasma	Aspartate Aminotransferase	mIU/mL	IU/L	microkat/L	1	0.0167
plasma	Aspartate Aminotransferase	microkat/L	IU/L	microkat/L	60	1
plasma	Cholesterol	mg/dL	mg/dL	mmol/L	1	0.0259
plasma	Cholesterol	mmol/L	mg/dL	mmol/L	38.67	1
plasma	Cholesterol	mmol/l	mg/dL	mmol/L	38.67	1

Step 5 - Handling character strings in collected Result

When the lab result is collected and coded, it may not be in a numeric representation and could result in character form as shown in Table 3.

Preservation of the original collected values in the data even when the collected value is being converted to our desired (conventional) units and standard units is required. The issue is - what would happen if the collected value contains character strings as shown below? The intent of the character string is context sensitive. In one context strings like those shown in rows 2 and 3 of Table 3 may be meant to convey that the data is categorical in nature, but that there is more content than just, say, 'high' and 'low'. Another context is that in which a string such as '<.01' is meant to convey the notion that a small amount was detected, and therefore the result is inconclusive.

Table 3:

LBTESTCD	LBORRES
ALT	< 7
ALT	> 8
CK	+15
CHOL	115

Again, this problem should be resolved through input from the statistician and clinical group. For this particular case, the statistician and clinical group may provide the following options:

Assigning values to those results shown in Table 4

Keep everything "as is" as nothing can be done

Preservation of collected results in the analysis dataset as they appeared in the original data is required, and it is wise to create another variable and do the manipulation according to the input received from the clinical group. This is illustrated in Table 4 where the new variable RESULTC, in most cases, is the same as the original lborres variable; note - some observations may have changed according to input from clinical/statistician.

Table 4:

LBTESTCD	LBORRES	RESULTC
AST	< 7	6.5
AST	> 8	9
CK	+ 15	15
CHOL	115	115

Step 6 - Finally, a numeric manipulation using the new variable, RESULTC follows along with upper limits and lower limits for the conversion. Careful attention should be given to character strings in the RESULTC variable when using the input statement. Please note that if you do not have character string attached to the "value," it is not necessary to create the RESULTC variable - use lborres directly. A few lines of code that address this are shown in the following:

```
data lab; set lab;
lborresn=input(resultc, best.);
lbornrhin=input(lbornrhi,best.);
lbornrlon=input(lbornrlo,best.);
run;
```

To carry the illustration forward, perform a PROC CONTENTS to show the way in which the original data has been preserved and its relation to RESULTC. Table 5 shows part of PROC CONTENTS for the illustrative lab data.

Table 5: An example of output from proc contents of the lab dataset:

Variable	Type	Len	Label
LBORNRHI	Char	200	Reference Range Upper Limit in Orig Unit
LBORNRLO	Char	200	Reference Range Lower Limit in Orig Unit
LBORRES	Char	200	Result or Finding in Original Units
lbornrhin	Num	8	*<- original upper limit numeric format*
lbornrlon	Num	8	*<- original limit numeric format*
lborresn	Num	8	*<- original result numeric format*
resultc	Char	200	*<- the new variable created after value was manipulated*

Step 7 - Import the spreadsheet to SAS® and use as source data for the conversion

In this example the dataset name is "convert."

Table 6: Proc contents of the source conversion dataset "convert":

Variable	Type	Len	Format	Label
cfactor	Num	8		cfactor
conventional_unit	Char	5	$5.	conventional unit
LBSPEC	Char	6	$6.	LBSPEC
LBTEST	Char	26	$26.	LBTEST
sfactor	Num	8		sfactor
standard_unit	Char	10	$10.	standard unit

Step 8 - Two datasets are in place - the source dataset that will be used for the conversion and the lab dataset - use a proc sql statement with left join to convert collected numeric values to conventional and standard units as illustrated below.

```
proc sql;
 create table  labconvt  as
 select a.*

    /* converting collected units to conventional units */

       a.lborresn*b.cfactor as LBCONN    ← conventional result
     , a.lbornrhin*b.cfactor as LBCONRHI   ← conventional ULN
     , a.lbornrlon*b.cfactor as LBCONRLO   ← conventional  LLN
      , b.conventional_unit as LBCONU    ← assigning conventional units
     /* converting collected units to standard units */
     , a.lborresn*b.sfactor as LBSTRESN    ← standard result
     , a.lbornrhin*b.sfactor as LBSTNRHI   ← standard ULN
     , a.lbornrlon*b.sfactor as LBSTNRLO   ← standard  LLN
      , b.standard_unit as LBSTRESU          ← assigning standard units
     from lab a left join convert  b
    on upcase(a.lbtest)=upcase(b.lbtest)
    and upcase(a.lbspec)=upcase(b.lbspec)
   and upcase(a.lborresu)=upcase(b.original_unit);
quit;
```

Due to the left join, the final dataset will have more observations than the original dataset.

FINAL OUTCOME

The main purpose of having different units in the datasets is to create an analysis lab dataset that can be used to facilitate the "one proc away" method of collapsing different units for use in programming the analysis tables, reports, and graphs for the Clinical Study Report.

The goal is to have a final Analysis dataset as illustrated below in Table 7:

Table 7:

USUBJID	LBTESTCD	UNITSYS	RESULT	UNITS	ULN	LLN
101	AST	ORIG	16	IU/L	8	22
101	AST	ORIG	1.5	IU/dL	.8	2.2
101	AST	ORIG	<8	mIU/mL	8	22
101	AST	ORIG	.4008	microkat/L	.1336	.3664
101	AST	CONV	16	IU/L	8	22
101	AST	CONV	15	IU/L	8	22
101	AST	CONV	7.5	IU/L	8	22
101	AST	CONV	24	IU/L	8	22
101	AST	SI	.2672	microkat/L	.1336	.3664
101	AST	SI	.25005	microkat/L	.1336	.3664
101	AST	SI	.1252	microkat/L	.1336	.3664
101	AST	SI	.4008	microkat/L	.1336	.3664

101	CHOL	ORIG	112	mg/dL	125	200
101	CHOL	ORIG	6.216	mmol/L	3.237	5.18
101	CHOL	ORIG	3.4965	mmol/l	3.237	5.18
101	CHOL	CONV	112	mg/dL	125	200
101	CHOL	CONV	240	mg/dL	125	200
101	CHOL	CONV	135	mg/dL	125	200
101	CHOL	SI	2.9008	mmol/L	3.237	5.18
101	CHOL	SI	6.216	mmol/L	3.237	5.18
101	CHOL	SI	3.4965	mmol/L	3.237	5.18

Since the creation of the actual analysis dataset is beyond the scope of this presentation, below is a brief explanation of how the data should be collapsed to create the final dataset.

For every single type of units, do a proc sort with nodupkey option. The code for the present illustration follows below: The variables in the "by" statement can be changed according to the project requirements.

```
proc sort data=labconvt  nodupkey out=orig; by usubjid lbspec lbtestcd visit lbdt lborres lborresu ;run;

proc sort data=labconvt nodupkey out=conv; by usubjid  lbspec lbtestcd visit lbdt lbconn lbconu ;run;

proc sort data=labconvt  nodupkey out=si; by usubjid lbspec lbtestcd visit lbdt lbstresn lbstresu ;run;
```

A required subsequent step is to append the three datasets and create "UNITSYS" variables to identify the unit system. It is required to change the formatting for the three lab results in order to collapse different units in the same columns.

CONCLUSION

1. The pharmaceutical industry is tightly regulated and must adhere to strict rules in the management and use of data in analyses. There is a need to preserve the original collected data "as is" without any manipulation. By doing so, an audit investigator will understand how it was done when looking at the analysis data and the LB domain.
2. Furthermore, by using the method presented in this paper, there is always a one-to-one mapping between the LB domain and the analysis data where unitsys ='ORIG.'
3. This source spreadsheet can be used across protocols and across studies.
4. The resulting dataset can be easily used to capture observations for the tables and listing using the "UNITSYS" variable for sub setting.
5. The desired display units in tables and listings can be changed as needed by the individual study teams.

REFERENCES

SAS Institute Inc. 2004, SQL Processing with SAS® Course Notes

ACKNOWLEDGMENTS

Thank you to Ram Radhakrishnan and Joyce Hernandeze for reviewing this work and providing constructive feedback.

AUTHOR CONTACT

Your comments and questions are valued and welcome. Contact the author at:
Pushpa Saranadasa
E-mail: Pushpa_saranadasa@merck.com