# Paper 182-2010
# Dealing with Lab Data – Stacking the Deck in Your Favor

Jennifer Fulton, Westat, Houston, TX

**Abstract**
In the world of clinical research, laboratory data yields some of the largest and most unwieldy data sets SAS programmers encounter in a project. Lab data usually is collected both on data collection forms (case report forms), and in an electronic format, such as a spreadsheet or text file straight from the lab. Extensive data manipulation is almost always required in order to get the data into a format that is useful for analysis, and to combine the other collected data with the data from labs correctly. These facts about lab data also mean that the odds of making a critical programming error are greatly increased.

This paper will provide a reference for the manipulation of typical lab data for the purpose of data analysis such as tables, listings, and graphs, as well as some time-saving tips. Sample SAS code is included throughout, which includes such manipulations as converting to standard units, dealing with normal ranges, assigning severity grades, and assigning Low, Normal, or High categories to values. Hopefully the paper can take some of the "labor" out of working with laboratory data!

**Manipulations of Lab Data – It Depends on the Hand You are Dealt**
Some or all of the following manipulations of the lab data are usually required to prepare the data to be used to create tables, listings, and graphs:

- Harmonize names of panels and analytes
- Create numeric values from character values
- Convert values and normal ranges to standardized units
- Convert WBC differentials into % or absolute counts
- Assign Low, Normal, or High category
- Assign Common Terminology Criteria (CTC) grades

Below is a discussion of the steps involved in completing each of the above updates. Each section also includes some example SAS code that can be inserted into a lab data program and expanded upon as necessary, depending upon the scope and content of the data provided and subsequent analysis required.

---

Harmonize names of panels and analytes
Lab data usually appears as a set of panels, and within each panel is a set of individual tests called analytes. Examples of panel names are Hematology, Chemistry, and Urinalysis. Examples of analyte names are Glucose, Potassium, and Red Blood Cell Count. There is not a regulatory standard for naming these panels or analytes, so if data comes in from more than one laboratory, one step the SAS programmer must complete is to "harmonize" the names of the panels and analytes so that the data sets can be appropriately mapped together. Generally, there are only a few panel names to deal with, and the project analyst may specify how panel names are to appear in titles in mock output, so harmonizing panel names is fairly straightforward. Individual analytes are another matter. One laboratory may abbreviate the lab analytes in the above example as "GLU", "K", "RBC". Another may concatenate the lab name with the units in parentheses such as "Glucose (mmol/L)", "Potassium (mmol/L)", RBC (10^12/L)". If a SAS programmer is faced with these different methods of naming analytes, he or she should create a new analyte name variable that is consistent for all labs.

Standardization of lab test names has been a goal for CDISC (Clinical Data Interchange Standards Consortium) in recent years. The creation of the LB SDTM (Study Data Tabulation Model) domain requires particular variables such as lab test identifiers that can take the guess work out of identifying an analyte. The reader can refer to the CDISC website for the complete list of CDISC controlled terminology. [1]

Units should be separated into a different variable in case conversion is necessary later, while taking care to assess whether the concatenated units are standardized or original units. Also if there is any question as to the meaning of an abbreviation, the SAS programmer should check with the analyst or the lab as to whether two different analyte names actually refer to the same thing. One final note: often in a single lab data set the SAS programmer may find 2 or more variables which are all ways of naming the analytes. For example, one variable may contain a very short, 3-letter abbreviation for every analyte, another may be a longer but still abbreviated version of the analyte name, and a third may spell out the analyte name in its longest form. It is wise to look at all of these variables to get a complete picture before attempting to harmonize analyte names.

Here are some common abbreviations for analytes:

| Abbreviation | Full Name |
|---|---|
| AlkPhos | Alkaline Phosphatase |
| ALT, SGPT | Alanine Aminotransferease |
| ANC | Absolute Neutrophil Count |
| APTT | Activated Partial ThromboplastinTime |
| AST, SGOT | Aspartate Aminotransferase |
| BUN | Blood Urea Nitrogen |
| CA | Calcium |
| CK | Creatine Kinase |
| FE | Iron |
| GGT | Gamma Glutamyl Transferase |
| iPTH | Intact Parathyroid Hormone |
| K | Potassium |
| NA | Sodium |
| PHOS | Phosphate or Phosphorus |
| PT | Prothrombin Time |
| RBC | Red Blood Cell Count |
| UA | Uric Acid |
| WBC | White Blood Cell Count |

Some analytes are almost always spelled out, such as Calcium, while others are almost always abbreviated, such as SGPT and SGOT.

Sample SAS code

```
/*Harmonize panel names
  Add to the list in parentheses if your data is different*/
LENGTH PANEL1 $50;
     IF PANEL IN ('U','URIN') THEN PANEL1 = 'URINALYSIS';
ELSE IF PANEL IN ('B','CHEM') THEN PANEL1 = 'CHEMISTRY';
ELSE IF PANEL IN ('H','HEMA') THEN PANEL1 = 'HEMATOLOGY';
:
:

/*Harmonize analyte names;
 Add to the list in parentheses if your data is different*/
LENGTH ANALYTE1 $50;
 IF ANALYTE IN ('ALK PHOS')               THEN ANALYTE1 = 'ALKALINE PHOSPHATASE';
ELSE IF ANALYTE IN ('ALAT (SGPT)','ALT')  THEN ANALYTE1 = 'ALT (SGPT)';
ELSE IF ANALYTE IN ('ASAT (SGOT)','AST')  THEN ANALYTE1 = 'AST (SGOT)';
ELSE IF ANALYTE IN ('CREAT CLR')          THEN ANALTYE1 = 'CREATININE CLEARANCE';
ELSE IF ANALYTE IN ('CHOL_TL')            THEN ANALTYE1 = 'TOTAL CHOLESTEROL';
ELSE IF ANALYTE IN ('KETONE')             THEN ANALYTE1 = 'KETONES';
:
:
```

Create numeric values from character values

Often one data set includes both a numeric and a character version of lab result values. In some cases these values are the same, such as a numeric 5 and a character '5'. But in other cases a numeric equivalent of the character value is not so obvious. Some examples include character values such as: 'NA', 'N/A', 'ND', '<2.0', 'Below Limit of Quantitation' (or 'BLQ'), 'Neg', 'Trace', and '+'. Usually for such values the numeric variable for this same record will have a missing value. Obviously numeric calculations that might be necessary to convert the original values to standard values cannot be performed on these character values as they are. Some analytes such as certain urine tests are not measured numerically so no conversion is expected (results are presented in frequency tables, not summary statistics). A clinician may have to make the final call as to whether or not the associated numeric value should always be treated as missing. For example, sometimes the decision is made to strip off any "<" or ">" and assign the remaining number as the numeric result. Or "BLQ" may be the same as "<0.7" for a particular lab machine, and this value may be analyzed as 0.7. The author recommends that any such adjustments be carefully documented by the SAS programmer.

Sample SAS code:

The following code would create a numeric version of a character variable, stripping off non-numeric characters.

```
if anyalpha(result_c)=0 then result_n = input(compress(result_c,'<>=+,.'),best12.);
```

For example, if result_c = '<3.00', the above code would derive result_n=3.

Convert values and normal ranges to standardized units

Different laboratories will have different equipment and calibrations for measuring lab result values. Thus, in order to combine data from more than one lab, the values must be converted into one standard unit for each analyte before values are summarized in a table or graph. Also if the normal ranges provided do not match the ranges designated as "standard normal ranges", (these will come either from one particular lab specified as the standard lab, or from other published, accepted documents), then the lab value must also be converted in a second step to adjust for the standard normal range. The most commonly used set of standard units is the International System of Units or SI. "All systems of weights and measures, metric and non-metric, are linked through a network of international agreements supporting the **International System of Units**. The International System is called the **SI**, using the first two initials of its French name *Système International d'Unités*. The key agreement is the Treaty of the Meter (*Convention du Mètre*), signed in Paris on May 20, 1875. 48 nations have now signed this treaty, including all the major industrialized countries. The United States is a charter member of this metric club, having signed the original document back in 1875." [2] This introduction to SI comes from the website http://www.unc.edu/~rowlett/units, which also contains a number of other pieces of information with regard to standardizing units in general (not just lab analytes). Another useful website with conversion calculators for SI as well as other units is http://medcalc3000.com.

Not all analysts will want units standardized based on the SI. Another set of known units is called US Units. Alternatively, one of the labs providing the lab data may be used as the "standard" (often the largest lab) and all other lab values must be converted to the units used by this lab. The author has been faced with varying levels of lab data information in past experience with clients. Following are example scenarios and SAS code to go with each instance.

**Scenario 1: A Full House**

Some clients provide a complete data set with variables containing original lab values, units, and normal ranges, and standardized lab values, units, and normal ranges (rare but wonderful!). In this case no work for conversion is necessary on the part of the SAS programmer. The SAS programmer can provide useful assistance in data cleaning by checking for outliers that may indicate that a value was not converted properly, and with other edit checks such as confirmation that each analyte has only one associated analyte name and standard unit. For some analytes the conversion factor and standard normal range are dependent on sex and/or age of the patient. Some normal ranges are also based on collection dates, as when the lab equipment was recalibrated in the middle of data collection. These factors must be taken into account in programming edit checks. It can also be useful to perform some longitudinal checks (the results for one patient across many visits, for example) along these same lines. Below is SAS code for one type of edit check.

Sample SAS code:

```
/*Edit check to watch out for disproportionate number of lows or highs*/
PROC SORT DATA = LABS;
BY LPARM;
RUN;

/*Count the number of Ls, Ns, and Hs*/
PROC FREQ DATA = LABS NOPRINT;
BY LPARM;
WHERE RNGFLAG NE ' ';
TABLE RNGFLAG/OUT=CHECK;
RUN;

/*Take a look at Ls and Hs that occur more than 75% of the time*/
DATA CHECK_BAD;
```

```
SET CHECK;
IF RNGFLAG IN ('L','H') AND PERCENT GE 75 ;
RUN;

DATA CHECK_FINAL;
MERGE CHECK_BAD(IN=A) CHECK;
BY LPARM;
IF A;
RUN;

PROC PRINT;
TITLE1 'USE THIS INFO TO CHECK DISPROPORTIONATE NUMBER OF Lows OR Highs';
RUN;
```

**Scenario 2: A Small Straight**

Some clients provide only original values and normal ranges, so that conversion of all values and normal ranges to the standard unit is up to the SAS programmer. Standardized units and conversion factors are usually provided in a separate file (data set or spreadsheet) that must be merged with the data that came straight from the lab. If the analyst simply says "Use SI" or provides a list of analytes and standard units, the SAS programmer will need to create this separate data set to merge in for standardizing. If the data comes from more than one laboratory, scenario 2 does not apply – standard normal ranges must be obtained, therefore scenario 3 (below) is a better fit.

Sample SAS code:

```
/*Converting one set of units at a time by multiplying the original
value and normal ranges by the conversion factor to get the standardized
value and normal ranges.*/
DATA LABS;
SET LABDATA;

/*conversion factor is 10 for g/dL to g/L*/
IF ORIG_UNIT = 'g/dL' AND STD_UNIT = 'g/L' THEN DO;
        STD_VALUE = ORIG_VALUE*10;
        STD_LO = ORIG_LOW*10;
        STD_HI = ORIG_HI*10;
END;
/*conversion factor is 1 for MEQ/L to mmol/L for most analytes*/
ELSE IF ORIG_UNIT = 'MEQ/L' AND STD_UNIT = 'mmol/L' THEN DO;
        STD_VALUE = ORIG_VALUE;
        STD_LO = ORIG_LOW;
        STD_HI = ORIG_HI;
END;
RUN;

/*Converting can be more efficient if the conversion factors are in a separate data set*/

/*After merging the data set containing the original values and normal ranges
with the data set containing standardized units and conversion factors,
multiply conversion factors in one data step*/
DATA LABS;
MERGE LABDATA(IN=A) CONVERSION_DATA;
BY ANALYTE ORIG_UNIT;
IF A;

/*WBC differentials cannot be converted this way so they are handled separately*/
IF ANALYTE NOT IN ('EOSINOPHILS', 'BASOPHILS'...) THEN DO;
        STD_VALUE = ORIG_VALUE*CONV_FACTOR;
        STD_LO = ORIG_LO*CONV_FACTOR;
        STD_HI = ORIG_HI*CONV_FACTOR;
END;
RUN;
```

**Scenario 3: A Few Cards Short of a Full Deck**
Sometimes the lab provides the original values and normal ranges, along with standard normal ranges, leaving only the conversion from original values to standardized values up to the SAS programmer. In this case the programmer can convert values to standard units and adjust for the standard normal range in one step by a basic transformation formula (see SAS code below).

One warning: With transformation it is possible for a value to become negative when standardized. Since a negative value is generally meaningless and physically impossible, it is usually set to 0. Such instances should be brought to the attention of the analyst for confirmation on how to deal with them.

Sample SAS code:

```
/*Transformation formula for standardizing units and ranges*/
/*Calculate the standardized value using the original normal range (LoOrig, HOrig)*/
/*and the standard normal range (LoStd, HiStd)*/
IF LoOrig NE . AND HiOrig NE . AND LoStd NE . AND HiStd NE . THEN
   STANDARDIZED_VALUE= LoStd +(((ORIG_VALUEN- LoOrig)/( HiOrig - LoOrig))*( HiStd - LoStd));
```

Note that order of operations and properly placed parentheses are key in obtaining an accurate result with this formula.

Some important general comments about conversion to standard units:

1. **For some analytes the conversion factor is dependent on the sex and/or age of the patient**, so code must take this into account. Age categories and other codes must match those in the standard dataset when combining to establish levels.
2. **Programmers are often faced with a mix of the levels of lab data described above**, meaning some labs can be converted with transformation and others require the use of conversion factors. Research and discussion with the analyst is often needed in order to obtain all of the necessary information to convert labs of interest. Transformation should be used when possible, then conversion factors if necessary.
3. **The conversion of WBC differentials is quite different from other lab analytes**. The transformation formula cannot be applied, and individual conversions are not as straightforward as for other lab analytes. Please see the next section of this document for the details.
4. **When performing conversion calculations, the SAS programmer should take great care with regard to the way decimal places are affected by the calculation**. The conversion factor may have more decimal places than the original value, and the analyst may want the standard values rounded to a specified level. Generally, the converted value should have at least as many significant digits as the standard normal range, though individual cases must be taken into account. The project statistician may be the person to consult.
5. **The conversion factor can differ from one analyte to another, even when converting from the same original to the same standard units**. Any conversion that involves moles (mol) is dependent on the molecular weight of the analyte. Since different molecules have different weights the conversion factors are different even though the units appear to be the same. Here are some examples: Uric Acid conversion factor for mg/dL to umol/L is 59.48, Bilirubin conversion factor for mg/dL to umol/L is 17.1, Creatinine conversion factor for mg/dL to umol/L is 88.4.
6. **It is recommended that an external document be used to house the standard unit conversion information**. It is advantageous to have that as a separate entity to allow you to utilize a 'lab expert' to review the spreadsheet. Such information can be stored in an Excel spreadsheet with such columns as "Analyte", "Original Unit", "Conversion factor", "Standardized Unit", and "Significant Digits". Once a standard spreadsheet is in place then one can create a macro to pull this information in that will convert the values as well as handle the number of significant digits preferred for reporting purposes.

<u>Convert WBC differentials into % or absolute counts</u>
White Blood Cells (WBC) are made up of a number of sub-components known as "differentials". WBC differentials may be reported as actual counts of the sub-component or as a percentage of the WBC count.

The analyst will make the decision as to whether counts or percentages will be the standard unit in which to report the differentials. Here is a list of the lab analytes that make up the WBC Differentials:

Eosinophils
Basophils
Lymphocytes
Monocytes
Neutrophils* (or "Total Neutrophils" or "Absolute Neutrophil Count")
-Band Neutrophils (or "Bands/Stabs" or "Bands")
-Segmented Neutrophils (or "Segs")
*Band Neutrophils + Segmented Neutrophils = Total Neutrophils

Converting WBC differentials from % to count or vice versa takes a somewhat different approach than the conversion method for other analytes.

**To convert WBC differentials from % (original unit) to counts (standard unit)**, divide the original value for the differential by 100 to make it a ratio instead of a percentage, and then multiply it by the standardized value for WBC count for the same subject at the same visit. The units for the standardized differential value will be the same as the units for WBC.

**To convert WBC differentials from counts (original unit) to % (standard unit)**, first make sure that the WBC counts and differential counts are in the same units (doesn't matter if this is original or standardized units). Then the differential percentage equals the differential value divided by the WBC value. Finally multiply the result by 100 to make it a percentage.

Sample SAS code:

```
/*Subset the WBC records*/
DATA WBC(KEEP=PT WBCVAL VISIT VISITDT VISITTM);
SET LABS;
WHERE ANALYTE = 'WBC';
RENAME SI_VALUEN = WBCVAL;
RUN;

PROC SORT DATA = WBC;
BY PT VISIT VISITDT VISITTM;
RUN;

/*WBCDIFFS contains the WBC differentials*/
PROC SORT DATA = WBCDIFFS;
BY PT VISIT VISITDT VISITTM;
RUN;

/*Merge WBCs with differentials by visit information*/
DATA WBCFIX(DROP=WBCVAL);
MERGE WBC WBCDIFFS;
BY PT VISIT VISITDT VISITTM;

*--From percentage to counts;
SI_VALUEN = WBCVAL*(ORIG_VALUEN/100);
SI_UNIT = '10^9/L';

      --- OR ---

*--From counts to percentages;
SI_VALUEN = (ORIG_VALUEN/WBCVAL)*100;
SI_UNIT = '%';
RUN;
```

Important note: These methods CANNOT be used to convert original normal ranges from % to counts or vice versa. In other words, it is not correct to standardize a normal range for a WBC differential by multiplying it by the standard normal range for WBC to go from % to counts, or vice versa dividing by the normal range for WBC to go from counts to %. That would make the differential normal range based on one measurement for one subject and that is not a normal range to apply to all subjects for all time points. If WBC differential values require conversion from counts to % or vice versa, the associated standard normal

ranges must be provided by the lab or some other source, or left blank. For the same reason, the transformation formula is not appropriate to apply to WBC differentials when converting between % and counts, even if both sets of normal ranges are provided.

Assign Low, Normal, or High category

Often the transition of a lab value from Low (L), Normal (N), or High (H) at baseline to L, N, or H at a follow-up visit is of interest for data analysis. Assigning these values is a straightforward matter of comparing the lab value to the associated normal range. If the value falls within the boundaries of the normal range, inclusive of the values, it is considered Normal. If the value falls below the lower boundary of normal it is considered "Low", and if the value falls above the upper boundary of normal it is considered "High". Missing lab values cannot be assigned such a category. Beware of the common programming error: IF VALUE LT LOWVAL THEN FLAG='L'. In this case missing values would be assigned to 'L' incorrectly. SAS programmers should also be careful to compare original values to original normal ranges only or standardized values to standard normal ranges only.

In rare cases a value may have one L, N, H value when using original values and normal ranges, but a different L, N, H value when using standardized values and normal ranges. For example, with transformation it is possible for an original value to become negative when standardized. Since a negative value is meaningless for most analytes, it would be set to 0, which might change it from Low to Normal if 0 is the lower bound of the normal range. Also, these flags can be different if a SAS programmer is instructed to blindly apply a standard set of normal ranges to all lab values without transformation. Specific cases of data with not enough information to easily assign the L, N, H values may have to be discussed with the analyst. Most of the time, only the "L" and "H" flags are actually displayed in the output ("N"s are left blank) to help draw attention to values that are outside normal ranges.

Sample SAS code:

```
/*Assign LNH flags*/
IF SI_VALUEN NE . THEN DO;
        IF .Z LT SI_VALUEN LT SI_LOW THEN LNH_FLAG = 'L';
        ELSE IF .Z LT SI_HIGH LT SI_VALUEN THEN LNH_FLAG = 'H';
        ELSE IF SI_LOW NE . AND SI_HIGH NE . AND
                SI_LOW LE SI_VALUEN LE SI_HIGH THEN LNH_FLAG = 'N';
END;
```

Assign Common Terminology Criteria (CTC) grades

CTC grades take the L, N, H assignments to the next level, providing more information about how *far* from normal a value actually is. Grades may be numeric such as 0 to 4, or character such as 'Mild', 'Moderate', 'Severe', or 'Life-threatening'. Assigning CTC grades can be a bit of an arduous process, especially since each analyte must be treated individually and direction (increase or decrease) plays a factor in the assignment as well.

Speaking very generally, grades are most often assigned by comparing the lab value to a range of values. The range may be specific (hard-coded) numbers, or it may be multiples of the low end or high end of the normal range. Some examples:

Example #1: For the analyte 'CD4' , the CTC grades are assigned as
0 if ">= 404 ";
2 if "< 404 and >=200";
3 if "< 200 and >=50";
4 if "< 50";
(for this particular lab a grade of 1 is not applicable).

Example #2: For the analyte 'TRIG' (Triglycerides), grades are assigned as
0 if "<= ULN";
1 if "> ULN";
2 if "> 2.5 x ULN and <=5.0xULN";
3 if "> 5.0 x ULN and <=10.0xULN ";
4 if "> 10.0xULN ";
where ULN = Upper Limit of Normal (from the normal range)

Again care should be taken not to accidentally assign grades to lab measurements that are missing.

The most frequently used rules for assigning severity grades are the CTC grades established by the National Cancer Institute (NCI), although they are often used for studies not related to oncology. The NCI CTC chart can be found on the website: http://ctep.info.nih.gov/reporting/ctc.html. The SAS programmer should also be sure to use the most current version of the NCI CTC chart, unless otherwise instructed by the analyst. As of the writing of this paper, this is version 3.0. SAS programs written to assign CTC grades should document the version number and source (NCI or otherwise) in the program header. If the programmer is asked for grades to be assigned to labs not covered by the CTC reference used, the analyst will have to provide further information such as their own grading system.

When using CTC grading systems, the SAS programmer should be careful that the lab values and the CTC grade ranges are in the same units, where applicable. Also sometimes the CTC grade ranges are dependent on combinations of specific (hardcoded) values and the normal ranges specified by the lab. In these cases it is possible for overlapping to occur when assigning grades. Here is an example:

> *Grades for Hypomagnesmia (Magnesium Increasing);
> 0="<= ULN";
> 1="> ULN and <=1.23";
> 2=" N/A";
> 3="> 1.23 and <= 3.30";
> 4="> 3.30";
> For the particular lab for one project, the Upper Limit of Normal (ULN) was 1.25. A value of 1.24 then could be assigned a grade of either 0 or 3 according to the grade definition above in combination with the specified ULN. And of course the definition for grade 1 is not even possible with this ULN and analyte.

Such cases must be handled with the analyst on a project and analyte-specific basis.

One last “challenge” that can arise with regard to CTC grades involves the direction of the abnormality. For most analytes the grades indicate how far below normal the lab value is OR how far above normal the lab value is. Only one direction is clinically important. But for a few analytes there are grades with respect to values above AND below the normal range. One example is Glucose. Too much glucose, known as hyperglycemia, or too little glucose, known as hypoglycemia, are both indications of clinical interest in analysis of lab data. So for glucose and other such labs there are two sets of CTC grades, often referred to as “increasing” and “decreasing”.

For these types of labs the most common approach to analysis is to assign TWO grades (one for “increasing” and one for “decreasing”) to every value. For example, a subject with a value of 40 for glucose would have a CTC grade of 2 for decreasing and 0 for increasing. Then two tables would be made to present “Glucose Increasing” and “Glucose Decreasing”. Such situations should be discussed in advance with the analyst to ensure the presentation of the results is meaningful and expected.

Sample SAS code:

```
/*Assign CTC grades*/
IF ANALYTE IN('GLUCOSE') AND SI_UNIT='mg/dL' THEN DO;
            * Grades for HYPERglycemia (increasing) ;
            IF SI_HIGH NE . THEN DO;
               GRDINC = 0 + (SI_VALUEN > SI_HIGH)
                          + (SI_VALUEN > MAX(SI_HIGH,8.9))
                          + (SI_VALUEN > MAX(SI_HIGH,13.9))
                          + (SI_VALUEN > MAX(SI_HIGH,27.8));
            END;
           * Grades for HYPOglycemia (decreasing);
            IF SI_LOW NE . THEN DO;
               GRDDEC = 0 + (SI_VALUEN < SI_LOW)
                          + (SI_VALUEN < MIN(SI_LOW,3.0))
                          + (SI_VALUEN < MIN(SI_LOW,2.2))
                          + (SI_VALUEN < MIN(SI_LOW,1.7));
            END;
 END;
```

**Cashing in Your Chips**

The items discussed in this paper attempt explain many of the difficulties faced with regard to lab data. Unfortunately lab data seems to be so variable depending on the source, that the SAS programmer should anticipate that odd values and records will probably appear that do not fit any of the "rules". Good edit checks and careful data review as well as frequent communication with the analyst are all keys to success when it comes to lab data. Good luck!

**References**

1. "CDISC Website, Laboratory Data Model " http://www.cdisc.org/content1058
2. "How Many? A Dictionary of Units of Measurement",
   © Russ Rowlett and the University of North Carolina at Chapel Hill, last revised Dec. 4, 2008,
   http://www.unc.edu/~rowlett/units/sipm.html

**Acknowledgments**

I would like to thank my supervisors who allowed and encouraged me to write and present this paper at SAS Global Forum. Special thanks to Duke Owen and Sherry Jones for your input, and to Stephen Black whose years of answering my questions about lab data lead to the writing of this paper.

**Contact Information**

Your comments and questions are valued and encouraged.

Jennifer Fulton, MS
Westat
5615 Kirby Drive, Suite 710
Houston, TX 77005
Work Phone: 713-353-7925
Fax: 713-529-4924
Email: JenniferFulton@westat.com



**Disclaimer**

The contents of this paper are the work of the author and do not necessarily represent the opinions, recommendations, or practices of Westat.