ABSTRACT

The competency of SAS datasets is based on the datasets' structure, content and relationship to other datasets. For the structure of 2 SAS data sets to be equivalent every element in the 2 sets must be the same. That is, each data set must be a subset of the other. Data editing for completeness and consistency can be described at 4 levels. Three levels exist within a SAS dataset, the fourth is between data sets. Internal integrity can be assessed with inter, intra variable checks as well as intra observation checks. Inter-dataset logical scans for structural consistency can be facilitated with the use of the contents procedure with an output out option. This four tiered approach to data editing is described with emphasis on structural integrity. Examples of the lack of structural integrity commonly encountered are presented. SAS code to perform structural integrity checks is also included.

INDIVIDUAL DATASET CONSIDERATIONS

In a real-world SAS environment there are always varying approaches to construction and maintenance of SAS code and SAS data sets. Programming experiences and backgrounds differ, work assignments change, applications require varying resources. Project management practices require operational terms and definitions to ensure coherence and integrity of SAS applications. What follows is an attempt to give a clear working definition of competent, complete and clean SAS data sets. In addition, the term "integrity" will be used to describe the relationship between 2 or more SAS datasets.

The level of detail of SAS datasets can vary by application. If we think of a SAS dataset as no more than a collection of SAS variables then a competent SAS dataset is no more than a collection of competent SAS variables. To be a competent SAS variable the variables attributes must meet the criteria below.

- The NAME assigned to the variable should be a meaningful acronym and/or be restricted to less than 8 characters to allow for concatenating characters in array or MACRO processing.
- The choice of variable TYPE (character or numeric) is important. Representation of numeric information must address exceptions by the use of special missing value codes. It may be necessary to differentiate between two missing value conditions, one where the data was never collected, the other where the data were collected but an invalid or no response was obtained.

Character variables can either be used as descriptors or for classification. Content of character variables used for classification is critical, whereas content of descriptors is less important. All of these issues must be addressed when deciding whether a SAS variable type is to be numeric or character.
- Explicit LENGTH declaration is required to avoid order-dependent or data-dependent processing results. Without explicit length declaration contents will dictate data structure, an obvious problem (Figure 3).
- LABELS provide the only opportunity in SAS to document a variable's content. Labels are two fold in purpose. They serve both to document a variable and to provide variable headers during report generation. It would be convenient if SAS provided a separate facility for variable documentation.

For report generation labels should (at SAS data set generation time) address print options and units of measure where applicable.
- FORMATS need to be stored as variable attributes to avoid redundancy at report generation.
A variable meeting all of the criteria above is said to be competent. A collection of competent SAS variables then will be a competent SAS dataset.

Further arguments about dataset competency can be made. It may be desirable for a SAS dataset to know its own creation date. Competency of a variable can be further enhanced if we allow a dictionary of legal values to be associated with a dataset. This dictionary not residing on the dataset but available to the dataset could be another attribute in the definition of a competent SAS variable.

What is a clean SAS dataset? Whether a dataset is clean is a content consideration, not a structure consideration. Data cleaning for content can be described at three levels.

The first level "a scrub" could be to run the dictionary of range checks against all variable values. Such scrubbing will need to incorporate missing values.

Secondly, intervariable checking considers the relationship 2 or more variables have with one another. Usually a new temporary variable is constructed to reflect the relationship between or among these variables. "Legal" values of this intervariable relationship are then "scrubbed" as in range checking above.

Lastly, before a SAS dataset is sufficiently scrubbed, inter-observation considerations must be made. In some cases the relationship between 2 class-variables must remain constant across observations. Figure 1. summarizes these three intra dataset scrubs.

Boundaries for the relationship among a set of continuous variables may be defined by the level of a classification. Data scrubbing here may consist of forming a statistic sensitive to aberrations in this relationship (Fig. 4).

Multiple Dataset Considerations

The relationship between 2 or more SAS datasets may dictate the structure of each.

Possibly it is helpful to think of any SAS variable as a collection of attributes, all of which must at least meet these minimal requirements (Fig. 2).

Figure 1. The four levels of checks required for competent SAS datasets.

SAS VARIABLE ATTRIBUTES

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>LABELS</th>
<th>FORMATS</th>
</tr>
</thead>
</table>

Figure 2. Minimum attributes common to all SAS variables. An additional attribute which would enhance a variable's competence is a library of valid ranges/legal values for each variable in a dataset.
For example, one dataset may be a subset of another [for merging, concatenating, etc.]. That is, one dataset may contain only some of the variables of the other. However, any variables common to both (or all) datasets must share identical variable attributes. Error detection at this interlevel dataset is structure based, not content based (Figure 1).

Two SAS datasets are said to lack structural integrity if they lack one-to-one variable attribute correspondence for variables common to each.

The code necessary to do content-checking is straightforward and easily incorporated into a menu-driven user-friendly system. For structural integrity checks, we have used the output option in the CONTENTS procedure to store the structural attributes of variables in a SAS dataset as the content of another SAS dataset. These attributes are then compared to identify potential problems. For illustration, we have the case where 2 datasets are to be identical. SAS code and results are given below (Figs. 5 & 6).

In conclusion, we have broken data editing into content-oriented and structure-oriented procedures. We have defined characteristics of each in such a way as to allow operational definitions of competency, cleanliness, and structural integrity in SAS datasets. SAS code was provided to demonstrate whether 2 or more structures created in a real-world working environment can claim structural integrity.
### Figure 5.
SAS code to perform the structural consistency checks between 3 data sets.

```sas
PROC CONTENTS DATA=L.AMPERSON OUT=AMPW_AMP骢2B ;
DATA _NULL_;SET AMPW;PRINT NAME TYPE LENGTH ;
RUN;

PROC COMPARE;
DATA WANTED;READ TST12345(IN=IN1);
READ TST22222(IN=IN2);
READ TST33333(IN=IN3);BY NAME TYPE LENGTH;
IF IN1=1 THEN TST12345="" ELSE TST12345="INCONSISTENT" ;
IF IN2=1 THEN TST22222="" ELSE TST22222="INCONSISTENT" ;
IF IN3=1 THEN TST33333="" ELSE TST33333="INCONSISTENT" ;
IF IN1=0 OR IN2=0 OR IN3=0 THEN FLAG="INCONSISTENT";
RUN;

PROC PRINT DATA=WANTED SPLIT='.';
VAR NAME TYPE LENGTH;
TITLE 'COMPARISON OF 3 DATA SETS';
RUN;

PROC COMPARE;
DATA TST12345 TST22222 TST33333;
RUN;
```

### Figure 6.
Output generated from structurally inconsistent data sets. NOTE: Data sets TST22222 and TST33333 are structurally consistent whereas TST12345 is inconsistent with TST22222 and TST33333.