Spring 2011 OASUS Questions and Answers

The following answers are provided to the benefit of the OASUS Users Group and are not meant to replace SAS Technical Support.

Question 1 : Using EG how do I delete specific rows from a dataset? For example, I want to delete from a dataset the rows with names= John Smith, Susan Toll, Vicky Jones, etc. Also, can I delete rows based on slight variations of these names?

The EG query builder does not allow you to delete rows but it allows you to select rows that you want to retain. The original dataset can then be overwritten with the selected rows.

If you don't mind writing SQL code yourself, you can use the DELETE statement to delete specific rows in one pass.

```
Proc sql;

    delete from cars2  WHERE Make = * 'oldsmobil';

quit;
```

In this particular example, all the rows where the Make column sounds like 'oldsmobil' will be deleted. You can also use a function such as SOUNDEX that a more sophisticated matching algorithm.

Question 2 : How can I get SAS to treat missing values as zeroes so I can perform mathematic operations?

Before answering the question, it is important to note that missing values have a meaning and in the majority of the cases they should be processed as such. SAS stores a missing value as a special numeric value. Therefore, it is only pertinent for numeric variables. Character variables use an empty string to represent a missing value.

Normally, the behavior of SAS functions and PROCs when processing missing values is quite acceptable. Lets SAS do its work! Some procedures do include options to influence the way missing are processed but in general missing values cannot be used for any computations since they don't represent valid numbers.

Often, SAS users think they can change the behavior by using the global MISSING option. This is not the case since the MISSING option simply allows you to select a different symbol to represent a missing value. By default the '.' is used.

If you want SAS to treat missing values as a specific value such as zeroes, the best way is to convert your data accordingly. Fortunately, SAS provides a very useful procedure called STDIZE to standardize numeric data.

```
/* Replace all missing values with 0s */
/* This is a permanent change*/

proc stdize data=miss reponly MISSING=0 out=nomiss;
var _numeric_;
run;
```

In this program all missing values in dataset MISS will be replace by zeroes in the output dataset NOMISS.

References :

SAS Global Forum Paper 275-2011, Why .1 + .1 Might Not Equal .2 and Other Pitfalls of Floating-Point Arithmetic. Clarke Thacher, SAS Institute Inc., Cary, NC

SAS Global Forum Paper 072-201, Special Missing Values for Character Fields. John Ladds, Statistics Canada, Ottawa, Ontario, Canada

Question 3 : How can I assign a user-defined format in a PROC SQL statement?

A SELECT statement in SQL contains a SELECT clause that is used to list the columns of interest. These columns will appear in the output (either in a report or a table). Along with the column name, you can specify a column modifier. This allows you to specify formats, labels and length.

Specifically, the following example shows you how to specify both a SAS format and a user-defined format.

```
/* Assign both a SAS format (8.) and */
/* a user-defined format (carsize) */

proc sql;

select invoice format=8.,cylinders format=carsize. as size
from sashelp.cars;

quit;
```

This example also uses an alias as a means to rename the column "cylinders" to "size".

Question 4 : How could I check the syntax of a DATA step without running the code?

The best way to do this is to use the CANCEL option on the RUN statement. By doing so, the step will not run but SAS will still validate the syntax.

The example below shows the use of the CANCEL option which can easily be turned on or off with a macro variable (&debug in the example).

```
%let debug=CANCEL;

/* Data step with syntax errors */
data new1;
     set sashelp.class;

     if (sex='M') then
          do;
               a=1;
               b=2    SYNTAX ERROR
               c=3;
          end;
run &debug;

/* Data step with no syntax errors */
data new2;
     set sashelp.class;

     if (sex='M') then
          do;
               b=2;
               c=3;
          end;
run &debug;
```

Note that the RUN CANCEL does not work when a DATA step reads in stream data using statements such as CARDS and DATALINES.

References :

SAS Global Forum Paper 097-2011, Conditional Processing Without Macros.
John Ladds, Statistics Canada, Ottawa, Ontario, Canada

Question 5 : What are the row and column limits for a SAS dataset?

The variables (columns) limits are dependent on the following factors:

1. *Version of SAS*
   In releases of SAS prior to SAS 9 except for Tru64 UNIX, the maximum number of variables in a single data set is 32,767.

2. *Observation length*
   An observation length cannot exceed 2GB on a 32-bit platform and $2**46$ on a 64-bit platform.

3. S*torage needed for names, labels, and other metadata*
   Assuming a single-byte character set, and that you use the maximum 352 bytes possible for name, label, and so on for each variable, you can have a maximum of about 4,050,000 variables. If the names, labels, and format names are shorter, you can have more than 200,000,000. There is a maximum of 1 GB to store all the variable names and other metadata (data set label, compression name, and so on).

Assuming the above limits are not exceeded, the maximum possible number of variables on either Windows or UNIX is 412,977,617 on 32-bit hosts and 2 GB on 64-bit hosts.

The row limits depends if you are running under a 32-bit platform (limit of 2 billions observations) and a 64-bit platform (limit of 9.2 quintillion observations) with the following Windows exceptions:

1. An observation cannot be longer than 5 MB. Therefore, in a data set that contains 32,767 character variables, the longest each variable can be is 160 bytes.

2. A SAS data set stored on a drive that is formatted as FAT16 has a file size limit of 2.1 GB.

3. A SAS data set stored on a drive that is formatted as FAT32 has a file size limit of 4 GB.

4. A SAS data set that is stored on an NTFS formatted drive has a file size limit of 2 TB.

5. A SAS data set might be limited to having $2**32-1$ observations. This limit can occur if the operating system is a 32-bit or 64-bit operating system.

In any cases, the resulting dataset must not exceed the operating system capabilities and the physical machine capabilities. For example, you may not have enough disk storage to store a data set.

If you have to deal with very large data sets and you're not sure if your current environment will handle them, you can contact SAS Technical support for advice. If you do indeed exceed the limits, you should review SAS Usage note 36112.

Question 6 : What are the recommended system requirements for SAS 9.2 and EG 4.3?

SAS has a number of documents that provide detailed requirements for installing and running SAS products on different computing platforms. In the case of this specific question, the following two documents would provide the necessary information:

- **System Requirements for SAS® 9.2 Foundation for Microsoft® Windows®**
- **System Requirements SAS® Enterprise Guide® 4.305**