PharmaSUG 2014 - Paper PO22

Tips for Finding Your Bugs Before QC Does

Beatriz Garcia, Inventiv Health Clinical Mexico, Mexico City

ABSTRACT

Statistical Programmers often have to work as both programmers and in validation or verification of programs during the course of a project. Sometimes after we deliver the program for validation, many requests are received in order to change the results or update the code. Why does that happen? Maybe because we have not checked the results as carefully as we should.

In this paper, if you work in UNIX environment, I want to show you some tips for reviewing your results before delivering to QC, so you can prevent many of those requests, avoid re-work and save time.

INTRODUCTION

When we start to work as Statistical Programmers and we finish our assignments, our Lead Project inform us that we have to send our programs to other programmers who were assigned to validate our results and documentation and oh surprise! We start to receive some emails or reports with all mistakes that we have done during our programming.

When you are working in UNIX environment and you don't have an easy access to SAS interactive, the editors are helpful for creating or editing the programs. In this paper I want to present some useful tips and SAS procedures for all programmers, from beginners to advanced, in order to avoid a headache when a program is submitted to QC/validation.

HEADER

A header helps to describe the purpose of the program, input, output short description, macros called and change history.

When a program is copied from another project, the person copying the program becomes the author,

So the first step: to review the header before sending our program to QC/validation, this can sound tedious but believe me, it is very helpful to be sure that it is complete and totally populated, i.e.:

```
***********************************************************************
* Program:        asl.sas
* Programmer:     Beatriz Garcia
* Date:           10DEC2013
* Purpose:        To create the ASL data set
* INPUT:
*   From:         rawdata: dm.sas7bdat
*                          vs.sas7bdat
* OUTPUT:
*   To:           outdata:..asl.sas7bdat
*
* Macros: %FixVar, %FixDate
* Changes: Bgarcia
*          30DEC2013 - Update the header for input data sets.
***********************************************************************;
```

The Changes section is very important to document the revisions made to the program. It should be updated each time the code is revised after finalization of the first version of a program.

PROC CONTENTS

Prints descriptions of the contents of one or more files from a SAS data library, i.e.

```
                             CONTENTS PROCEDURE
Data Set Name: WORK.ORANGES                       Observations:         4
Member Type:   DATA                               Variables:            5
Engine:        V9                                 Indexes:              0
Created:       15:56 Monday, April 27, 1999       Observation Length:  40
Last Modified: 15:56 Monday, April 27, 1999       Deleted Observations: 0
Protection:                                       Compressed:          NO
Data Set Type:                                    Sorted:             YES
Label:

                 -----Engine/Host Dependent Information-----

       Data Set Page Size:         6144
       Number of Data Set Pages:   1
       First Data Page:            1
       Max Obs per Page:           152
       Obs in First Data Page:     4
       Number of Data Set Repairs: 0
       Physical Name:              SYS96050.T153830.RA000.USERID.R0000004
       Release Created:            8.0000B1
       Release Last Modified:      8.0000B1
       Created by:                 USERID
       Last Modified by:           USERID
       Subextents:                 1
       Total Blocks Used:          1
                      Taste Test Results For Oranges

                           CONTENTS PROCEDURE

          -----Alphabetic List of Variables and Attributes-----

                     #    Variable    Type    Len    Pos
                     -----------------------------------
                     2    FLAVOR      Num       8      8
                     4    LOOKS       Num       8     24
                     3    TEXTURE     Num       8     16
                     5    TOTAL       Num       8     32
                     1    VARIETY     Char      8      0
```

The procedure output provides values for the physical characteristics of the SAS data set WORK.ORANGES. Here we have the important values:

Observations	Is the number of nondeleted records in the data set.
Observation Length	Is the maximum record size in bytes.
Compressed	Has the value NO if records are not compressed; it has the value CHAR or BINARY if records are compressed.
Data Set Page Size	Is the size of pages in the data set.
Number of Data Set Pages	is the total number of pages in the data set.
First Data Page	is the number of the page that contains the first data record; header records are stored in front of data records.
Max Obs per Page	Is the maximum number of records a page can hold.
Obs in First Data Page	Is the number of data records in the first data page.

Talking about deriveds, you should review the following aspects:

- Attributes: Label, length, type, format
- Check proc contents variables vs. specifications in order to include only those variables needed.
- If there are variables without label before sending our output to QC.

PROC PRINT

For reviewing your data partially, is recommended to use PROC PRINT to see how data is stored in the data set.

SYNTAX

PROC PRINT *<option(s)>*;

BY <DESCENDING> *variable-1* <...<DESCENDING> *variable-n*>
<NOTSORTED>;

PAGEBY *BY-variable*;

SUMBY *BY-variable*;

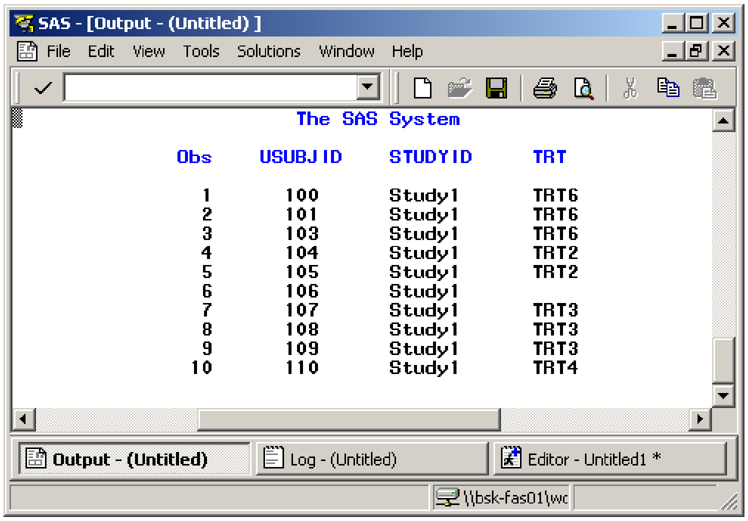
ID *variable(s) <option>*;

SUM *variable(s) <option>*;

VAR *variable(s) <option>*;

Although it is a tool very complete, we can simplify it i.e.

```
PROC PRINT data=ASL (obs=10)
        Var Usubjid Studyid trt;
Run;
```



```
                   The SAS System

            Obs    USUBJID    STUDYID    TRT

              1      100      Study1     TRT6
              2      101      Study1     TRT6
              3      103      Study1     TRT6
              4      104      Study1     TRT2
              5      105      Study1     TRT2
              6      106      Study1
              7      107      Study1     TRT3
              8      108      Study1     TRT3
              9      109      Study1     TRT3
             10      110      Study1     TRT4
```

In this example, one of the patients has an empty TRT, in this case, you can back to the code and check what is happening and fix it.

PROC FREQ

Use a PROC FREQ to check counts, completeness and consistency of data, since it provides descriptive statistics about a particular data set and it is very helpful for analyzing healthcare data sets, especially on TLG's.

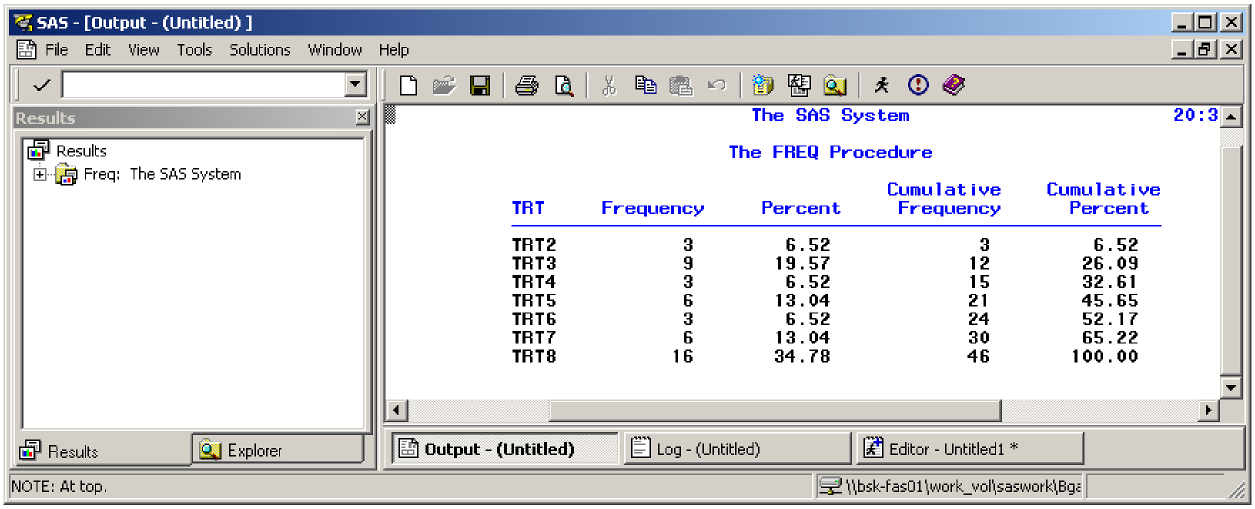
Although it has a very huge capacity, it can be used in its simple form, only for reviewing the numbers on how many patients should be for each treatment, how many patients are male or female, etc.

The basic syntax for one-way table is:

```
PROC FREQ <options>;
TABLES variable 1 * variable 2;
Run;
```

i.e.

```
PROC FREQ Data=ASL;
          Tables trt;
Run;
```



```
                          The SAS System                          20:3

                        The FREQ Procedure

                                          Cumulative    Cumulative
  TRT        Frequency     Percent         Frequency      Percent
  ------------------------------------------------------------------
  TRT2              3        6.52                 3         6.52
  TRT3              9       19.57                12        26.09
  TRT4              3        6.52                15        32.61
  TRT5              6       13.04                21        45.65
  TRT6              3        6.52                24        52.17
  TRT7              6       13.04                30        65.22
  TRT8             16       34.78                46       100.00
```

NOTE: At top.

In the example we can see the frequency of TRT variable and how many patients we have in the whole Data set.

PROC MEANS

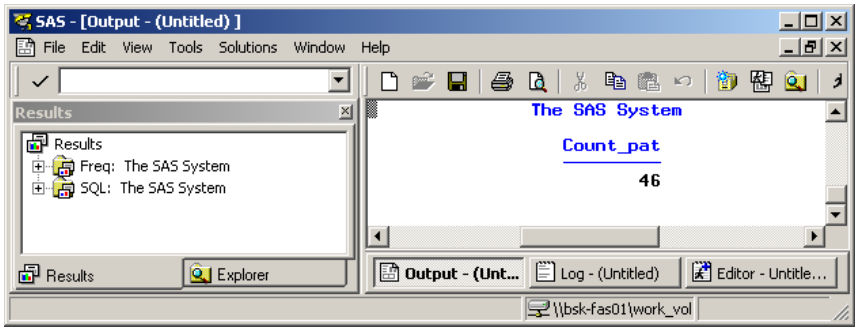
Use PROC MEANS for analyzing the values of numeric variables

```
PROC MEANS Data=ASL n mean median min max;
Title "Demographics";
  var age weigth;
  by trt;
run;
```

PROC SQL

Use PROC SQL in its simple form for simplifying the code and get the total of patients.

```
PROC SQL;
  select count( distinct usubjid) as Count_pat from ASL;
 quit;
```

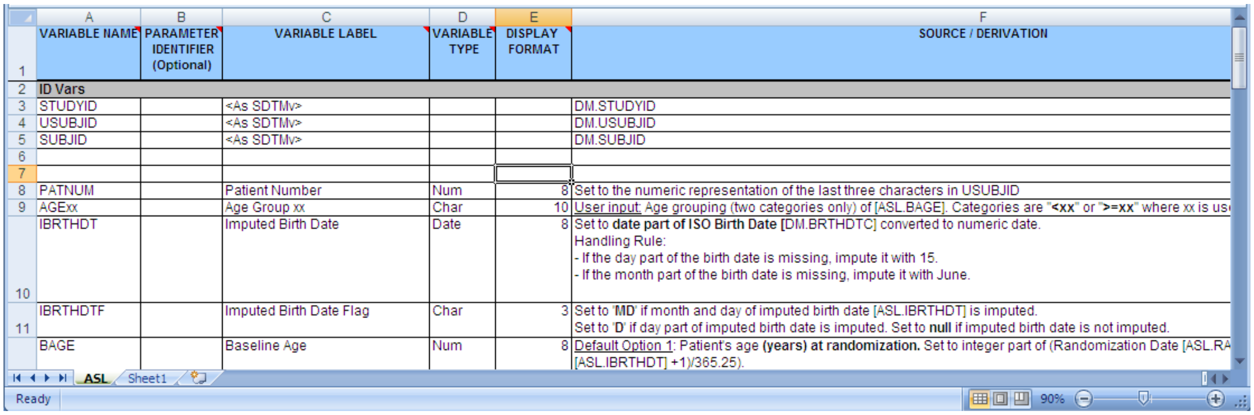


```
The SAS System

  Count_pat

         46
```

SPECS

Specs is commonly known as Specifications for programming Deriveds, Tables, Listings or Graphs and it is the document that shows the rules on how to do the output.

So the best is to be sure that this document is including all the variables and their derivations for creating the outputs properly.



	A	B	C	D	E	F
1	VARIABLE NAME	PARAMETER IDENTIFIER (Optional)	VARIABLE LABEL	VARIABLE TYPE	DISPLAY FORMAT	SOURCE / DERIVATION
2	ID Vars					
3	STUDYID		<As SDTMv>			DM.STUDYID
4	USUBJID		<As SDTMv>			DM.USUBJID
5	SUBJID		<As SDTMv>			DM.SUBJID
6						
7						
8	PATNUM		Patient Number	Num	8	Set to the numeric representation of the last three characters in USUBJID
9	AGExx		Age Group xx	Char	10	User input: Age grouping (two categories only) of [ASL.BAGE]. Categories are "**<xx**" or "**>=xx**" where xx is us
10	IBRTHDT		Imputed Birth Date	Date	8	Set to **date part of ISO Birth Date [**DM.BRTHDTC] converted to numeric date. Handling Rule: - If the day part of the birth date is missing, impute it with 15. - If the month part of the birth date is missing, impute it with June.
11	IBRTHDTF		Imputed Birth Date Flag	Char	3	Set to '**MD**' if month and day of imputed birth date [ASL.IBRTHDT] is imputed. Set to '**D**' if day part of imputed birth date is imputed. Set to **null** if imputed birth date is not imputed.
	BAGE		Baseline Age	Num	8	Default Option 1: Patient's age **(years) at randomization.** Set to integer part of (Randomization Date [ASL.RA [ASL.IBRTHDT] +1)/365.25).

Once the outputs are ready, pay special attention to titles and footnotes and do the comparison between the output and the specifications, ensure that all the variables are included.

CONCLUSION

We need to be sure that the header is populated correctly and we need to provide the complete information since the programs are property of the companies and someone else can be in charge of the same code and the correct header can avoid rework or spent extra time, the documentation is very important.

It's a very good suggestion to add these Procedures to our code in order to prove that we have checked our results.

Using simple techniques we can review our results before sending to QC or validation and avoid a couple of bad stripes to our performance.

RECOMMENDED READING

- Base SAS® Procedures Guide
- http://support.sas.com/onlinedoc/912/docMainpage.jsp

ACKNOWLEDGEMENTS

Thanks to Nancy Brucken, Elizabeth Grant, Israel León, Nadia Cárdenas and Alberto Hernández for their support, knowledge and patience.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Beatriz Garcia
Enterprise: Inventiv Health Mexico
Address: San Francisco #1005 Del Valle, Benito Juarez,
City, State ZIP: Mexico City, 03100
Work Phone: +52 55 5827 0917
E-mail: beatriz.garcia@inventivhealth.com