

Paper 122-29

To REPORT or to TABULATE? – That is the Question!

Dan Bruns, Tennessee Valley Authority, Chattanooga, TN
Ray Pass, Ray Pass Consulting, Hartsdale, NY

ABSTRACT

Should I use PROC REPORT or PROC TABULATE to produce that report? Which one will give me the control and flexibility to produce the report exactly the way I want it to look? Which one is easier to use? Which one is more powerful? **WHICH ONE IS BETTER?**

If you have these and other questions about the pros and cons of the REPORT and TABULATE procedures, this presentation is for you. We will discuss, using real-life report scenarios, the strengths (and even a few weaknesses) of the two most powerful reporting procedures in SAS (as we see it). We will provide you with the wisdom you need to make that sometimes difficult decision about which procedure to use to get the report you really want and need.

INTRODUCTION

Well, here we are again “titans” fans due to the incredible persuasion of one of the Hands On Workshop chairs, Deb Cassidy. NOT to do “battle” as we have done the previous three SUGIs, but rather to turn our knowledge of these two incredibly **simple** and **powerful** procedures into a workshop.

We will basically be using some basic code you will be submitting to help you discover the power of these two “titan” procedures. We will walk through a scenario using one procedure to produce a report then produce the same report with the other procedure. Then we will take the same code and expand upon it in the next scenario. Hopefully you will be enlightened as to where each procedure’s strengths lie. So you can reproduce and experiment with the examples, we chose to use the SHOES dataset from the SASHELP library. We chose this one because we felt it poses many of the same attributes of some typical datasets/database tables most reports are generated from (categorical values and actual/summarized values). Another set of datasets in SASHELP that you might find useful to experiment with are the ones that begin with PRD.

#	Variable	Type	Len	Pos	Format	Informat	Label
6	Inventory	Num	8	16	DOLLAR12.	DOLLAR12.	Total Inventory
2	Product	Char	14	57			
1	Region	Char	25	32			
7	Returns	Num	8	24	DOLLAR12.	DOLLAR12.	Total Returns
5	Sales	Num	8	8	DOLLAR12.	DOLLAR12.	Total Sales
4	Stores	Num	8	0			Number of Stores
3	Subsidiary	Char	12	71			

We have not included any code here as we have not quite decided which of the many useful features to cover during the presentation and each scenario. The final product will be on the SUGI website. But, in each of the scenario descriptions below we will discuss things (in general) that will be presented.

For those of you who may not know the underlying design premise for these two procedures, we have included the following excerpt from one of our earlier papers that was written for us by Alan Eaton of SAS Institute, the original developer of BOTH TABULATE and REPORT.

The TABULATE Procedure

Hierarchies Are Good. The primary goals for PROC TABULATE were to form and present hierarchical tables of summary statistics using the data values of classification variables to define hierarchies. An important model for TABULATE was TPL, a table producing language developed at the Department of Labor. TPL did a wonderful job of defining, building, and presenting complex hierarchical tables. However, TPL required users to predefine a significant amount of structural information in codebooks and could require as many as 15 JCL steps to execute. Our goal was to make the table building process entirely data driven and to simplify the task to writing a few lines of SAS code rather than many lines of JCL.

Give Me the Summary Version. PROC SUMMARY already computed summary statistics. TABULATE’s job was to organize these summary statistics for presentation. Other SAS procedures were available in the early days for detailed listings, PROC PRINT, and row and column computations, PROC COMPUTAB. TABULATE made no attempt to offer these kinds of features.

Generality in All Dimensions. PROC TABULATE supports three-dimensional tables, tables with pages, rows, and

columns. TABULATE was designed to support arbitrarily complex nestings and concatenations of table elements in each dimension. So TABULATE provides a lot of flexibility but at the cost of seeming overly complex and difficult to master at times.

The REPORT Procedure

Hierarchies Are Not So Good. The syntax for TABULATE is a challenge for a lot of users. Some form of simplification was in order. The approach taken in PROC REPORT was to limit the number of dimensions and to support a simpler COLUMN statement. Even that COLUMN statement had to be complex enough to support nestings of ACROSS variables and other elements.

Give Me the Facts. It was also necessary to offer detail reports with more formatting options than PROC PRINT afforded. PROC REPORT supports both GROUP and ORDER variables to generate both summary and detail reports.

Hierarchies Are Not So Good. Users frequently requested that TABULATE support a “post compute” feature to allow new rows or columns to be calculated from the rows and columns of summary statistics already computed from the data. How to specify symbols for use in expressions was a major stumbling block in TABULATE. The expression terms could be very complicated specifications of a sub-dimension of the TABLE statement. So the generality of TABULATE worked against this opportunity for enhancement. Supporting computed columns was a requirement of REPORT from the beginning. The simpler COLUMN statement made it easier to identify symbols to use in column expressions. With symbols available for parsing it became possible to build an interface between PROC REPORT and the DATA step so that PROC REPORT users can compute new columns using DATA step language and functions.

SCENARIO #1

Since a lot of reports these days are mostly a summary of the data (in our humble opinion), that is what we will be focusing on. If you really need to do a detail report of all the data, REPORT or PRINT can do that but TABULATE is really not designed for that.

To even the playing field a little to begin with, we chose a data set that basically had its data already in summarized fashion. In all honesty, TABULATE is NOT meant to be an item lister, unless the items already represent data summaries, as they do in SHOES (total numbers per Region/Subsidiary/Product).

In this scenario we will produce a simple straightforward report. We will discuss the basic statements needed/required for each procedure and their function: for TABULATE, CLASS, VAR, and TABLE; for REPORT, COLUMN and DEFINE; What happens by default; the concepts of concatenation and nesting.

SCENARIO #2

In this scenario we will expand upon the previous code to enhance the look of the report. In other words, what in the previous report would you change to make it more presentable? What options/features can we add to the already used statements to improve the appearance of the report? In this scenario we will be seeing how to add labels, titles, and formats to the report to enhance the aesthetics.

SCENARIO #3

In this scenario we will expand upon the previous code even further to enhance the report. Topics discussed: Subtotaling; Let's add some additional/different statistics (i.e. MEAN, MEDIAN, MAX, etc.); how can we control the order the data is presented; how are missing values treated/controlled; and some others we are thinking about. Additional statements discussed: for TABULATE, KEYLABEL and additional TABLE, CLASS & VAR options; for REPORT, COMPUTE and BREAK and additional DEFINE options.

SCENARIO #4

In this scenario we will expand upon the previous code even further to discuss simple percentages and a few other things (surprises?) we have not thought of**yet!**

IN SUMMARY

“So, which is better?”, you might ask. Well, our task was not to declare a winner and a loser, but rather to show that either procedure could be used to solve your report problem, but each has a definite set of strengths over the other depending on the desired result.

This paper is not intended to be a cure for all your TABULATE and REPORT problems. Every use is unique in some way. All we have attempted to do is give you a head start, and perhaps coax you to further discover the power that

these two SAS “TITANS” possess (to be clear, we’re talking about the **procedures** here, **not** the authors). The tools are there and they are powerful foundations to valuable and productive report generation.

CONTACT INFORMATION

Your comments and questions are always valued and encouraged. Please contact the authors at:

Dan Bruns
Tennessee Valley Authority
1101 Market Street (MP 2B)
Chattanooga, TN 37402
423/751-6430 Fax 423/751-3163
debruns@tva.gov

Ray Pass
Ray Pass Consulting
5 Sinclair Place
Hartsdale, NY 10530
914-693-5553 Fax 914-206-3780
raypass@att.net

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.