

Battle of the Titans: REPORT vs TABULATE

Dan Bruns, Tennessee Valley Authority, Chattanooga, TN
 Ray Pass, Ray Pass Consulting, Hartsdale, NY
 Alan Eaton, SAS Inc., Cary, NC

ABSTRACT

Are you in a quandary about whether to use PROC REPORT or PROC TABULATE for a certain report? Are you unsure which one will give you the control and flexibility to produce the report the way YOU want it to look? If you have these and other questions about the pros and cons of REPORT and TABULATE, this presentation is for you! We will discuss, using actual examples, the strengths (and even a few weaknesses) of two of the most powerful reporting procedures in the SAS System, thus providing with you the wisdom to help you make that sometimes difficult decision of which procedure to use to get the report you really want and need.

IN THE BEGINNING (AE)

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.

HOW IT ALL BEGAN (DB, RP)

Some years ago (more than we care to recall), DB and RP were sitting around at some SUGI discussing the strengths and benefits of the then fairly new procedures REPORT and TABULATE, both of which were created by AE. We both used both procedures, but RP quickly became known as Dr. Report and DB became rather infamous as Mr. Tabulate. Several years later we were having another discussion about the two most used reporting procedures in the SAS System and how users seem to be confused about which one to use. So, we came up with the idea of a paper highlighting the strengths and purpose of each procedure. To add a little spice to the show, we decided to turn it into a "battle".

The ground rules of the battle are simple. We take four scenarios of different kinds of reports and solve them with both REPORT and TABULATE. 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. As an added treat we have included a solution for each scenario using ODS STYLES to illustrate the true power of the procedures.

As the code and resulting outputs quickly become voluminous, we have not included them here in the written version of this "paper". They are however available as an attachment to this paper on the CD version of the NESUG '01 Conference Proceedings as well as on the NESUG website (www.nesug.org) under Conference Proceedings (NESUG '01).

As this paper is NOT meant to be a tutorial in REPORT, TABULATE, ODS or STYLES, there is no explanation given of the code. Instructional material is available in numerous sources including SAS and BBU publications, SUGI and Regional SUG conference Proceedings, SAS-L, and many online tutorials produced by various educational and commercial institutions.

With that in mind ...

LET THE GAMES BEGIN !

Scenario #1: Simple Detail

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.) With that caveat in mind, both procedures can produce simple detail listings, as can be seen in the results (**Simple Detail – REPORT** and **Simple-Detail – TABULATE**). REPORT does a better job with column headings and summary row separators. These minor deficiencies are much less obvious when ODS and STYLES are used to enhance the outputs (**Simple Detail - REPORT - ODS (styles)** and **Simple Detail - TABULATE - ODS (styles)**). In all of the ODS/STYLES code and results presented in this paper, the specific style items used are only one of an almost infinite set of possible configurations. These are only meant to be a springboard. The possibilities are staggering!

Scenario #2: Simple Summary

Both REPORT and TABULATE can produce simple statistics with minimal effort as seen in the results (**Simple Summary – REPORT** and **Simple Summary – TABULATE**). Both resultant tables present their results in a clear and productive manner. Once again, as can be seen in the ODS/STYLES outputs (**Simple Summary – REPORT – ODS (styles)** and **Simple Summary – TABULATE – ODS (styles)**), the results can be significantly visually enhanced. It is worth repeating the fact that the STYLES configurations used here are only one of an almost endless number of possibilities.

Scenario #3: “Stats on the side”

In this scenario, the task was to simply flip the statistics from columns to rows. This is really a piece of cake for TABULATE as can be seen in the code. It is also doable in REPORT, but does require some pre-processing of the data somewhere before REPORT takes over. The solution presented is perhaps only one of many that could prepare the data for REPORT to do the job. As with the previous scenario, both procedures can produce quite usable outputs in a vanilla fashion (**“Stats on the Side” – REPORT** and **“Stats on the Side” – TABULATE**) or an ODS/STYLES enhanced version (**“Stats on the Side” – REPORT – ODS (styles)** and **“Stats on the Side” – TABULATE – ODS (styles)**).

Scenario #4: “Variables-On-the-Fly”

Here, the task was to produce new variables for reporting which did not exist in the original data set. REPORT excels in its ability to produce transient “computed” variables with almost the entire DATA step collection of data manipulation tools available to it. With some creative coding, TABULATE can also get the job done, although with more complex manipulation called for (or even more simple manipulations such as a simple subtraction of one variable from another) TABULATE would have had to resort to pre-processing as REPORT did in the previous scenario. Nevertheless, as can be seen in the simple results (**“Variables On-the-Fly” – REPORT** and **“Variables-On-the-Fly” - TABULATE**), both procedures can accomplish the task. As was the case in each of the previous scenarios, the addition of ODS/STYLES adds significantly to the usability of the reports in terms of visual presentation (**“Variables On-the-Fly” – REPORT – ODS (styles)** and **“Variables-On-the-Fly” – TABULATE – ODS (styles)**). As an added bonus, “traffic-lighting” was added to the output of both REPORT and TABULATE to highlight this capability.

IN SUMMARY

“So, who won?”, 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 even the most complex report problem (sometimes with a lot of extra work), 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). We also wanted to demonstrate that when you wed REPORT or TABULATE to ODS and STYLES, the results are far greater than the mere addition of a few lines of code. The tools are there and they are powerful foundations to valuable and productive report generation.

ACKNOWLEDGEMENTS

We would like to say a big “Thank You!” to Sandy McNeil (current REPORT developer) and David Kelley (current TABULATE developer) of SAS for their support and participation in this project.

SAS is a registered trademark of the SAS Institute, Inc., Cary, NC.

AUTHORS

If you have any questions or comments, please write or call:

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

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