The Battle of the Titans (Part II):
PROC REPORT versus PROC TABULATE

Kirk Paul Lafler, sasNerd, Spring Valley, CA
Joshua M. Horstman, Nested Loop Consulting, Indianapolis, IN
Ben Cochran, The Bedford Group, Raleigh, NC
Ray Pass, Gainfully Retired and Having the Time of His Life, St Louis, MO
Dan Bruns, Very Happily Retired

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 knowledge you need to make that difficult decision about which procedure to use to get the report you really want and need.

Introduction
Well, here we go again “Titans” fans – due to the incredible persuasion of the “ORIGINAL” Titan champions, Ray Pass and Dan Bruns, three new warriors, Kirk, Joshua, and Ben, have united to do battle again using the colossus procedures: PROC REPORT and PROC TABULATE. NOT to replicate the “battle” as was waged at previous SAS® Users Group International (SUGIs) conferences, but rather to turn our knowledge of these two incredibly powerful procedures into a “fun”, and engaging presentation.

We will basically be using some basic code to help you discover the power of these two “titan” procedures. We will walk through several scenarios 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 CARS dataset from the SASHELP library. We chose this dataset because we felt it possessed many of the same attributes of some typical datasets/database tables most reports are generated from (categorical values and actual/summarized values).

Data Set Used in Examples
The examples used throughout this battle of the titans utilizes the dataset, SASHELP.CARS. The SASHELP.CARS dataset consists of 428 observations and 15 variables, illustrated below.
Our focus for this paper and presentation is to explore an assortment of scenarios describing general concepts that users will most likely encounter while using PROC TABULATE and PROC REPORT. For users who may not know the underlying design premise for these two procedures, we have included the following excerpt from earlier papers that were written by Dan Bruns, Ray Pass and Alan Eaton of SAS Institute, the original developer of both the REPORT and TABULATE procedures.

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.
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 complex code.

**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.

**Round #1**

Many users produce detail output and reports on their data, we will begin here. (We will examine how a summary report can be produced in Round #2). In this round, we will examine the REPORT and TABULATE procedures to produce detail output / reports. Several statements and options are available to users of PROC REPORT and PROC TABULATE to produce detail output / reports. For the REPORT procedure, the COLUMN(S) and DEFINE statements can be specified; and for the TABULATE procedure, the CLASS, VAR, and TABLE statements can be specified; see below.

**PROC REPORT Code:**

```sas
TITLE1 Detailed Cars Report <= $13,000 ;
TITLE2 Arranged by Origin ;
TITLE3 Produced by PROC REPORT ;
PROC REPORT DATA=sashelp.cars ;
COLUMNS origin type make model MSRP ;
DEFINE origin / ORDER 'Origin of Car' WIDTH=6 ;
DEFINE type  / ORDER 'Type of Car' WIDTH=8 CENTER ;
DEFINE make  / ORDER 'Make of Car' WIDTH=13 ;
DEFINE model / DISPLAY 'Model of Car' WIDTH=40 ;
DEFINE MSRP  / DISPLAY WIDTH=8 FORMAT=Dollar8. ;
WHERE MSRP LE 13000 ;
RUN ;
```

**PROC TABULATE Code:**

```sas
TITLE1 Detailed Cars Report <= $13,000 ;
TITLE2 Arranged by Origin ;
TITLE3 Produced by PROC TABULATE ;
PROC TABULATE DATA=sashelp.cars ;
CLASS origin type make model ;
VAR MSRP ;
TABLES MSRP ;
WHERE MSRP LE 13000 ;
RUN ;
```
Results:

![Detailed Cars Report <= $13,000 Arranged by Origin Produced by PROC REPORT](image)

PROC TABULATE Code:

```
TITLE1 Detailed Cars Report <= $13,000 ;
TITLE2 Arranged by Origin ;
TITLE3 Produced by PROC TABULATE ;
PROC TABULATE DATA=sashelp.cars FORMAT=DOLLAR8. ;
CLASS origin type make model ;
TABLE origin * type * 
    make * model, 
    MSRP ;
VAR MSRP ;
WHERE MSRP LE 13000 ;
RUN ;
```
Results:

Round #2
In the second round we will turn our attention from the production of detail output / reports to summary output / reports. In other words, we will describe and illustrate the specific statements and options to produce summary-level output / reports that display several descriptive statistics (i.e., SUM, N, MEAN, MIN, and MAX); using PROC REPORT and PROC TABULATE.

PROC REPORT Code:

```
TITLE1 MSRP Summary Statistics Report ;
TITLE2 Cars <= $18,000 ;
TITLE3 Produced by PROC REPORT ;
PROC REPORT DATA=sashelp.cars ;
COLUMNS type make MSRP MSRP=MSRP_Sum MSRP=MSRP_N MSRP=MSRP_Mean MSRP=MSRP_Min MSRP=MSRP_Max ;
DEFINE type / GROUP 'Type of Car'  WIDTH=8 ;
DEFINE make / GROUP 'Make of Car'  WIDTH=13 ;
DEFINE MSRP / ANALYSIS NOPRINT FORMAT=Dollar12. ;
DEFINE MSRP_SUM / ANALYSIS SUM FORMAT=Dollar12. "Sum" ;
DEFINE MSRP_N / ANALYSIS N "N" FORMAT=COMMA7. ;
DEFINE MSRP_MEAN / ANALYSIS MEAN "Mean" ;
```
DEFINE MSRP_MIN / ANALYSIS MIN "Min";
DEFINE MSRP_MAX / ANALYSIS MAX "Max";
WHERE MSRP LE 18000;
RUN;

Results:

PROC TABULATE Code:

TITLE1 MSRP Summary Statistics Report;
TITLE2 Cars <= $18,000;
TITLE3 Produced by PROC TABULATE;
PROC TABULATE DATA=sashelp.cars;
CLASS type make;
TABLE type * make,
    MSRP * (SUM N MEAN MIN MAX);
VAR MSRP;
WHERE MSRP LE 18000;
RUN;
Results:

Round #3
In the third round we will use PROC REPORT and PROC TABULATE, along with the Output Delivery System (ODS) Excel destination, to send output / reports to a Microsoft® Excel spreadsheet. We describe and illustrate the SAS-supplied style templates, the ODS Excel destination, and options, below.

PROC TEMPLATE List Styles Code:

```
PROC TEMPLATE;
  LIST STYLES;
RUN;
```

The results produced from the PROC TEMPLATE LIST STYLES statement produces the following styles, see below.
Results:

The next example illustrates the creation of an Excel spreadsheet by specifying the ODS Excel destination with PROC REPORT.

**PROC REPORT Code:**

```plaintext
ODS Excel FILE='REPORT-to-Excel.XLSX' STYLE=BarrettsBlue ;
TITLE1 Detailed Cars Report <= $13,000 ;
TITLE2 Arranged by Origin ;
TITLE3 Produced by PROC REPORT ;
PROC REPORT DATA=sashelp.cars ;
   COLUMNS origin type make model MSRP ;
   DEFINE origin / ORDER 'Origin of Car' WIDTH=6 ;
   DEFINE type / ORDER 'Type of Car' WIDTH=8 CENTER ;
   DEFINE make / ORDER 'Make of Car' WIDTH=13 ;
   DEFINE model / DISPLAY 'Model of Car' WIDTH=40 ;
   DEFINE MSRP / DISPLAY WIDTH=8 FORMAT=Dollar8. ;
   WHERE MSRP LE 13000 ;
RUN ;
ODS Excel CLOSE ;
```
Results:

The next example illustrates the creation of an Excel spreadsheet by specifying the ODS Excel destination with PROC TABULATE.

**PROC TABULATE Code:**

```sas
ODS Excel FILE='TABULATE-to-Excel.XLSX'
STYLE=BarrettsBlue ;
TITLE1 Detailed Cars Report <= $13,000 ;
TITLE2 Arranged by Origin ;
TITLE3 Produced by PROC TABULATE ;
PROC TABULATE DATA=sashelp.cars FORMAT=DOLLAR8. ;
CLASS origin type make model ;
TABLE origin * type *
   make * model,
   MSRP ;
VAR MSRP ;
WHERE MSRP LE 13000 ;
RUN ;
ODS Excel CLOSE ;
```
Results:

![Table of vehicle models and their MSRP prices]

**Round #4**

In the fourth, and final, round we will illustrate various style options with PROC REPORT and PROC TABULATE code to enhance the look of the output / report. What and how would you change to the code to make it more presentable? What statements, options and features could be specified to improve the appearance of the output / report? In the following table several style components are illustrated using PROC REPORT and PROC TABULATE, see below.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>Affects the report and the table structure.</td>
</tr>
<tr>
<td>Header (HDR)</td>
<td>Affects color header cells.</td>
</tr>
<tr>
<td>Column</td>
<td>Affects data cells.</td>
</tr>
<tr>
<td>Summary</td>
<td>Affects summary lines generated by BREAK or RBREAK statements.</td>
</tr>
<tr>
<td>Lines</td>
<td>Affects lines generated by LINE statements.</td>
</tr>
</tbody>
</table>

Specific style attributes available to PROC REPORT and PROC TABULATE are illustrated, see below.
### Style Attribute Description

<table>
<thead>
<tr>
<th>Style Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Background</strong>=</td>
<td>Changes background color (e.g., Red, Blue, Green, Orange, Yellow).</td>
</tr>
<tr>
<td><strong>Bordercolor</strong>=</td>
<td>Changes border color (e.g., Red, Blue, Green, Orange, Yellow).</td>
</tr>
<tr>
<td><strong>Borderwidth</strong>=</td>
<td>Changes border width (e.g., 0, 3, 7, in, cm, pt).</td>
</tr>
<tr>
<td><strong>Cellspacing</strong>=</td>
<td>Changes cell spacing (e.g., 0, 3, 7, in, cm, pt).</td>
</tr>
<tr>
<td><strong>Foreground</strong>=</td>
<td>Changes foreground color (e.g., Red, Blue, Green, Orange, Yellow).</td>
</tr>
<tr>
<td><strong>Frame</strong>=</td>
<td>Changes frame style (e.g., ABOVE, BELOW, BOX, HSIDES, LHS, RHS, VOID and VSIDES).</td>
</tr>
<tr>
<td><strong>Rules</strong>=</td>
<td>Changes rules (e.g., ALL, COLS, GROUPS, NONE, and ROWS).</td>
</tr>
</tbody>
</table>

In the next PROC REPORT code we examine how labels, titles, formats, and the style= option can be specified to enhance the output / report aesthetics.

**PROC REPORT Code:**

```plaintext
TITLE1 Customized Cars Report <= $13,000 ;
TITLE2 with Style ;
TITLE3 Produced by PROC REPORT ;
PROC REPORT DATA=sashelp.cars ;
   COLUMNS origin type make model MSRP ;
   DEFINE origin / ORDER 'Origin of Car' WIDTH=6 ;
   DEFINE type  / ORDER 'Type of Car' WIDTH=8 CENTER ;
   DEFINE make  / ORDER 'Make of Car'  WIDTH=13 ;
   DEFINE model / DISPLAY 'Model of Car' WIDTH=40 ;
   DEFINE MSRP  / DISPLAY WIDTH=8 FORMAT=Dollar8.
   STYLE(Column)={Background=Red
                  Foreground=White} ;
   WHERE MSRP LE 13000 ;
RUN ;
```
Results:

![Customized Cars Report <= $13,000 with Style](image)

In the next PROC TABULATE code we examine how labels, titles, formats, and the style= option can be specified to enhance the output / report aesthetics.

PROC TABULATE Code:

```plaintext
TITLE1 Customized Cars Report <= $13,000 ;
TITLE2 with Style ;
TITLE3 Produced by PROC TABULATE ;
PROC TABULATE DATA=sashelp.cars FORMAT=DOLLAR8. ;
CLASS origin type make model ;
TABLE origin * type *
  make * model,
  MSRP * {style={Background=red
          Foreground=White}} ;
VAR MSRP ;
WHERE MSRP LE 13000 ;
RUN ;
```
Results:

In the next PROC REPORT code we examine how labels, titles, formats, and the style= option can be specified to enhance the output / report aesthetics by displaying traffic lighting scenarios as a “background” color.

PROC REPORT Code:

TITLE1 Customized Cars Report ;
TITLE2 PROC REPORT with Traffic Lighting ;
PROC FORMAT ;
    VALUE MSRPFmt LOW - < 20000 = 'Green'
        20000 - < 30000 = 'Blue'
        30000 - < 40000 = 'Orange'
        40000 - HIGH = 'Red' ;
run ;
PROC REPORT DATA=SASHELP.Cars
    STYLE(Header)={BackGround=Blue ForeGround=White
                   Font=(Arial, 10pt, Bold)} ;
    COLUMNS Origin Make Type Model MSRP ;
    DEFINE Origin / ORDER 'Country of Origin' WIDTH=6 ;
    DEFINE Make / ORDER 'Make of Vehicle' WIDTH=13 ;
    DEFINE Type / ORDER 'Vehicle Type' WIDTH=8 ;
    DEFINE Model / DISPLAY 'Vehicle Model' WIDTH=40 ;
DEFINE MSRP / ANALYSIS 'Vehicle MSRP' WIDTH=8
   STYLE(Column)=[FontWeight=Bold
                  ForeGround=White
                  BackGround=MSRPFmt.] ;
RUN ;

Results:

In the next PROC TABULATE code we examine how labels, titles, formats, and the style= option can be specified to enhance the output / report aesthetics by displaying traffic lighting scenarios.

PROC TABULATE Code:

TITLE1 Customized Cars Report;
TITLE2 PROC TABULATE with Traffic Lighting;
PROC FORMAT;
   VALUE MSRPFmt LOW - < 20000 = 'Green'
       20000 - < 30000 = 'Blue'
       30000 - < 40000 = 'Orange'
       40000 - HIGH = 'Red';
run ;
PROC TABULATE DATA=SASHelp.Cars
FORMAT=DOLLAR10.
STYLE={BackGround=Blue ForeGround=White
Font=(Arial, 10pt, Bold)} ;
CLASS origin type make model / Style={BackGround=Blue
ForeGround=White} ;
TABLE origin * type * make * model,
MSRP * {STYLE=[FontWeight=Bold
ForeGround=White
BackGround=MSRFmt.]} ;
VAR MSRP ;
RUN ;

Results:

<table>
<thead>
<tr>
<th>Origin</th>
<th>Type</th>
<th>Make</th>
<th>Model</th>
<th>MSRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>Hybrid</td>
<td>Honda</td>
<td>Civic Hybrid 4dr manual (gas/electric)</td>
<td>$20,140</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Insight 2dr (gas/electric)</td>
<td>$10,110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Toyota</td>
<td>Prius 4dr (gas/electric)</td>
<td>$20,610</td>
</tr>
<tr>
<td>SUV</td>
<td></td>
<td>Acura</td>
<td>MDX</td>
<td>$36,845</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Honda</td>
<td>CR-V LX</td>
<td>$19,860</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Element LX</td>
<td>$18,690</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pilot LX</td>
<td>$27,560</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hyundai</td>
<td>Santa Fe GL S</td>
<td>$21,599</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Isuzu</td>
<td>Ascender S</td>
<td>$31,649</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rodeo S</td>
<td>$20,449</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kia</td>
<td>Sorento LX</td>
<td>$19,635</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lexus</td>
<td>GX 470</td>
<td>$45,700</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LX 470</td>
<td>$44,800</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RX 330</td>
<td>$39,190</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mazda</td>
<td>Tribute DX 2.0</td>
<td>$21,887</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mitsubishi</td>
<td>Endeavor XLS</td>
<td>$30,493</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Montero XLS</td>
<td>$33,112</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Outlander LS</td>
<td>$18,892</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nissan</td>
<td>Pathfinder Armada SE</td>
<td>$33,840</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pathfinder SE</td>
<td>$27,339</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Xterra XE V6</td>
<td>$20,939</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Suzuki</td>
<td>Vitara LX</td>
<td>$17,163</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XL-7 EX</td>
<td>$23,690</td>
</tr>
</tbody>
</table>
**Conclusion**

So, “Which colossus procedure is better?”, you might ask. Well, our task was not to declare a winner or a loser, but rather to show that PROC REPORT and PROC TABULATE could be used to solve your reporting needs. This paper / presentation is intended to illustrate the subtle differences between PROC REPORT and PROC TABULATE. What we have attempted to do is give you a head start, and hopefully ignite an excitement for further discovery about these two SAS “TITANS” procedures. (To be clear, we’re talking about the procedures here, not the authors.) It is also worth mentioning that each procedure has a definite set of strengths over the other depending on the desired result. The tools are there and they serve as powerful foundations to valuable and productive report generation.

**References**


**Acknowledgments**

The authors thank the PharmaSUG 2023 Conference Committee, particularly the Solution Development Section Chairs, Ajay Gupta and Renuka Tammisetti, for accepting our abstract and paper; the PharmaSUG 2023 Executive Committee for organizing and supporting a “live” conference event; SAS Institute Inc. for providing SAS users with wonderful software; and SAS users everywhere for being the nicest people anywhere!

**Trademark Citations**

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.
About the Authors

Kirk Paul Lafler is a SAS, SQL, Python, Excel, and data analytics educator, developer, programmer, and consultant; a lecturer and adjunct professor at San Diego State University (SDSU); an adjunct professor at University of California San Diego (UCSD) Extension; and teaches SAS, SQL, Python, Excel, and data analytics courses, workshops, and webinars to users around the world. Kirk has been a SAS user since 1979 and is the author of several books including, PROC SQL: Beyond the Basics Using SAS, Third Edition (SAS Press. 2019) along with papers and articles on a variety of SAS topics. Kirk has participated as an Invited speaker, educator, keynote, and section leader at SAS conferences and education forums; and is the recipient of 27 “Best” contributed paper, hands-on workshop (HOW), and poster awards.

Joshua M. Horstman is an independent statistical programmer with 18 years’ experience using SAS in the pharmaceutical industry. He has experience working for GlaxoSmithKline, Eli Lilly and Company, inVentiv Health, and others. Josh has presented numerous papers at PharmaSUG, SAS Global Forum, and other SAS User Group meetings. He holds a bachelor’s degree in mathematics and computer science, and a master’s degree in statistics from Colorado State University.

After more than 11 years with SAS Institute in the Professional Services (as an Instructor) and Marketing Departments (as Marketing Manager for the SAS/EIS product), Ben Cochran left to start his own consulting and SAS Training business in the fall of 1996 – The Bedford Group. As an affiliate member of SAS Institute’s Alliance Partner Program, Ben has been involved in many teaching and consulting projects over the last 20+ years. Ben has authored and presented several papers at SUGI, SGF, and regional user groups on a variety of topics since 1988.

After 35 years or so, Ray Pass, one of the original champions of the Battle of the Titans, retired on June 1, 2016 to be exact, (but who cares) from full-time SASing. He now spends his time living, thinking, reading, cooking, eating, walking, biking, shopping, erranding, and as much flaneuring as he can. After recently coming out of retirement he is having the time of his life solving SAS puzzles, teaming up with a great bunch of people with lots of mutual respect and getting paid for it - couldn't be better. He has adopted a new life motto: “Every day is Saturday!”

Dan Bruns – A legendary SAS user for many years and one of the original champions of Battle of the Titans.

Comments and suggestions can be sent to:

Kirk Paul Lafler
SAS® / SQL / Python / Excel / Data Analytics Educator, Developer, Programmer, Consultant, Data Analyst, and Author
sasNerd
E-mail: KirkLafler@cs.com
LinkedIn: http://www.linkedin.com/in/KirkPaulLafler
Twitter: @sasNerd

Joshua M. Horstman
Senior SAS® Consultant, Application Developer and Author
E-mail: jmhorstman@gmail.com
LinkedIn: https://www.linkedin.com/in/joshuahorstman

Ben Cochran
SAS® Consultant, Application Developer, Programmer and Educator
The Bedford Group
E-mail: bentcochran@gmail.com
LinkedIn: https://www.linkedin.com/in/ben-cochran-35577112/

Dr. Ray Pass
SAS® Consultant, Programmer, Data Analyst and Author
“Retired and Having the Time of His Life”
E-mail: RayPass@att.net
LinkedIn: https://www.linkedin.com/in/ray-pass-phd-152781b/
Dan Bruns
SAS® Programmer, Data Analyst and Author
“Very Happily Retired”
E-mail: danielebruns@gmail.com