

What's Next?

Thomas J. Winn, Jr.

Texas State Auditor's Office, Austin, Texas

Abstract

Intended for attendees who have learned the material that was covered in the earlier presentations in the "Introduction to SAS" Section, this short paper concludes the Section by describing some recommendations for the next developmental steps which could be taken by beginning SAS programmers. The paper includes highlights of the following topics: writing customized reports (PROC PRINT, DATA _NULL_, PROC TABULATE, & PROC REPORT), the SAS macro language, and PROC SQL.

Recommendations for your next developmental steps

Let us assume that one has learned the material which was covered in the earlier presentations in the SSU 2001 "Introduction to SAS" Section. Now, I would like to conclude this Section by describing some recommendations regarding the next developmental steps that could be taken by beginning SAS programmers.

Specifically, I will include highlights of the following topics:

- ◆ Output Delivery System,
- ◆ customized report writing, using PROC PRINT, PROC REPORT, PROC TABULATE, and DATA _NULL_,
- ◆ the SAS macro language, and
- ◆ PROC SQL.

A working knowledge of these topics should be included in the basic toolkit of every SAS programmer.

Here are a few ways to take those next steps:

- ◆ Take a training class,
- ◆ Attend tutorial presentations at SAS users conferences,
- ◆ Consult with an expert,
- ◆ Read the documentation or other appropriate reference materials.

Most of the topics are covered in papers presented in the SSU 2001 Tutorials section!

Output Delivery System

Beginning with Version 7, the SAS System provided an ability to deliver procedure output in a flexible variety of file types and formats, through the SAS Output Delivery System (ODS). ODS combines raw data with table definitions to produce output objects, which can be sent to one or more ODS destinations. Currently, the available ODS destinations include: the Listing destination, the Output destination (SAS Data Set), the HTML destination, the Printer destination (PostScript, PDF, PCL), and the RTF destination.

ODS destinations can be either open or closed. When a destination is open, ODS can send output objects to it. And whenever an ODS destination is closed, output

objects cannot be sent to it. ODS statements are used to control different features of the Output Delivery System. For example, they could be used to open, close, or manage an ODS destination. Moreover, they can be used anywhere in a SAS program.

Here is an example of the use of ODS statements to direct some SAS output to HTML:

```
ods html file='odshtml-body.htm'  
      contents='odshtml-contents.htm'  
      page='odshtml-page.htm'  
      frame='odshtml-frame.htm';  
proc univariate data=sashelp.class;  
  var height;  
  id name;  
  title 'Descriptive Statistics Concerning Height  
in the Sashelp.Class Dataset';  
run;  
ods html close;
```

And here is a portion of the HTML output that was generated:

The screenshot shows the SAS Output window with the title "Descriptive Statistics Concerning Height in the Sashelp.Class Dataset". Below the title, it says "The UNIVARIATE Procedure Variable: Height". There are two tables displayed. The first table, titled "Moments", has columns for various statistical measures and their values. The second table, titled "Basic Statistical Measures", has columns for Location and Variability measures and their values.

Moments			
N	19	Sum Weights	19
Mean	62.3368421	Sum Observations	1184.4
Std Deviation	5.12707525	Variance	26.2869006
Skewness	-0.2596696	Kurtosis	-0.1389692
Uncorrected SS	74304.92	Corrected SS	473.164211
Coeff Variation	8.22479143	Std Error Mean	1.17623173

Basic Statistical Measures			
Location		Variability	
Mean	62.33684	Std Deviation	5.12708
Median	62.80000	Variance	26.28690
Mode	62.50000	Range	20.70000

In most cases, the default output style will be adequate. However, the programmer may customize ODS output by specifying style definitions, which affect the colors, font, size, etc.

Customized Reports with PROC PRINT

PROC PRINT has several features and options which can be used to customize the appearance of detail reports printed with this procedure.

- ◆ DATA=, DOUBLE, HEADING=, LABEL, N, NOOBS, OBS=, ROUND, SPLIT=, and UNIFORM,
- ◆ VAR, ID, BY, SUM, PAGEBY, SUMBY, FORMAT, TITLE, and FOOTNOTE.

PROC TABULATE

PROC TABULATE is used to build tabular reports containing descriptive statistical information, including

hierarchical relationships among variables. PROC TABULATE is particularly well-suited for preparing summary reports (where each row represents multiple observations). However, it is less well-suited for producing detail reports (single row for each observation).

Here is some SAS code for a simple tabular report:

```
libname company 'C:\Program Files\SAS
Institute\SAS\VB\core\sample';
proc tabulate data=company.empinfo
format=5.0;
class division gender;
keylabel n= '
all='Total';
label division = 'Division'
gender = 'Gender';
table division all, gender all / rts=30
misstext='0';
title 'Summary of Employee Information';
run;
```

Here is the associated report:

Division	Gender		Total
	F	M	
ADMINISTRATION	1	2	3
CALIFORNIA REGIONAL	5	12	18
CONTRACTS	3	1	4
CORPORATE COMMUNICATIONS	3	5	8
DOCUMENTATION DEVELOPMENT	2	15	17
EXECUTIVE	6	3	9
FACILITY MAINT	8	13	21
FINANCE	1	3	4
HOST SYSTEMS DEVELOPMENT	11	11	22
HUMAN RESOURCES	7	18	25
INFORMATION SYSTEMS	7	10	17
INTERNAL DATA BASE	0	4	4
PUBLICATIONS	7	11	18
QUALITY ASSURANCE	10	10	20
SALES & MARKETING	11	21	32
SOFTWARE DEVELOPMENT	13	35	48
TECHNICAL SUPPORT	6	14	20
TEXAS REGIONAL	7	15	22
VIDEO	1	2	3
Total	104	295	399

PROC REPORT

PROC REPORT is a flexible procedure that can produce a variety of reports. PROC REPORT offers more control and customization than PROC PRINT. PROC REPORT can be used to produce both detail and summary reports; however, it is not as good as PROC TABULATE for producing hierarchical tables.

Here is some SAS code for a simple PROC REPORT step:

```
proc report data=sashelp.shoes headskip
headline missing;
column subsidiary product sales;
define subsidiary / order 'Subsidiary'
format=$15.;
define product / order 'Product'
format=$15.;
define sales / analysis sum 'Sales'
format=dollar15.2;
break after subsidiary / ol summarize skip
suppress;
title 'Product Sales According to Subsidiary';
run;
```

Here is the report which was produced:

Subsidiary	Product	Sales
Africa	Boat	\$29,161.00
	Men's Casual	\$67,242.00
	Men's Dress	\$26,710.00
	Sandal	\$62,813.00
	Slipper	\$68,841.00
Asia	Sport Shoe	\$1,230.00
	Women's Casual	\$51,541.00
	Women's Dress	\$108,942.00
Australia	Boat	\$15,462.00
	Men's Casual	\$249,201.00
	Men's Dress	\$221,445.00
	Sandal	\$1,280.00
	Slipper	\$48,750.00
Europe	Sport Shoe	\$449.00
	Women's Casual	\$225,516.00
	Women's Dress	\$241,829.00
Japan	Boat	\$1,153,667.00
	Men's Casual	\$21,237.00
	Men's Dress	\$63,266.00
	Sandal	\$123,713.00
	Slipper	\$29,138.00
North America	Sport Shoe	\$64,891.00
	Women's Casual	\$2,617.00
	Women's Dress	\$39,145.00
South America	Boat	\$29,141.00
	Men's Dress	\$4,932.00
	Sandal	\$23,000.00
	Sport Shoe	\$5,117.00
	Slipper	\$19,160.00
Total	Boat	\$42,769.00

Writing Customized Reports

FILE and PUT statements can be used in a DATA _NULL_ step to write special reports according to detailed specifications. The disadvantage is that pointer controls must be used to specify the placement of every item in the report.

The SAS Macro Language

The SAS Macro Language is a powerful programming tool ...

- ◆ for simplifying repetitive coding,
- ◆ for communicating information between program steps,
- ◆ for generating data-dependent SAS statements,
- ◆ for permitting conditional execution of SAS code, and
- ◆ for dynamically importing certain information from the SAS Supervisor.

Macros are stored text that contain entire blocks of SAS code, and which are identified by a name. The stored text can include SAS statements, literals, numbers, macro variables, macro functions, macro expressions, or calls to other macros.

Macro variables are used to facilitate symbolic substitution of strings of text, whereas macros can be used to manipulate SAS source statements. Macro information can be inserted at any point in a SAS program simply by referring to the macro entity by name, preceded by a special character, which distinguishes macro statements from ordinary SAS code.

PROC SQL

Structured Query Language (SQL) is a language that talks to a relational database management system. PROC SQL processes SQL statements that read and update tables. Besides being useful for queries, it also is a powerful tool for data manipulation.

PROC SQL uses Structured Query Language to . . .

- ◆ retrieve and manipulate SAS data sets,
- ◆ create and delete data sets,

- ◆ add or modify data values in a data set,
- ◆ add, modify, or drop columns in a data set,
- ◆ create and delete indexes on columns in a data set.

Components of the SAS System

Do you remember this display from the very first "Introduction to SAS" presentation? It is a listing of many of the components of the SAS System.

- Base SAS	- SAS/LAB
- SAS/ACCESS	- SAS/MDDDB Server
- SAS/AF	- SAS/OR
- SAS/ASSIST	- SAS/QC
- SAS/CONNECT	- SAS/SHARE
- SAS/EIS	- SAS/SPECTRAVIEW
- SAS/ETS	- SAS/STAT
- SAS/FSP	- SAS/TOOLKIT
- SAS/GIS	- SAS/AppDev Studio
- SAS/GRAPH	- SAS/Enterprise Guide
- SAS/IML	- SAS/Enterprise Miner
- SAS/INSIGHT	- SAS Universal ODBC Driver
- SAS/IntrNet	- SAS/Warehouse Administrator

After becoming familiar with Base SAS, novice SAS programmers might begin learning other pertinent components of the SAS System.

"What's Next?" Summary

Build on the foundation of knowledge you already have acquired. Your next developmental steps might include the following suggested topics:

- ◆ Output Delivery System,
- ◆ customized report writing, using PROC PRINT, PROC REPORT, PROC TABULATE, and DATA _NULL_,
- ◆ the SAS macro language,
- ◆ PROC SQL,
- ◆ other components of the SAS System.

Here are some ways by which you might take those next developmental steps:

- ◆ Take a training class.
- ◆ Attend tutorial presentations at SAS users conferences (Be sure to review the many excellent papers in the SSU 2001 Tutorials Section!).
- ◆ Consult with an expert.
- ◆ Read the documentation or other appropriate reference materials.

The next step is up to you.

Suggested References:

SAS Institute Inc., [SAS OnlineDoc, Version 8](#)
 SAS Institute Inc., [Getting Started With the SAS System, Version 8](#)
 SAS Institute Inc., [SAS Language Reference: Concepts, Version 8](#)
 SAS Institute Inc., [SAS Language Reference: Dictionary, Version 8, Volumes 1 and 2](#)
 SAS Institute Inc., [SAS Procedures Guide, Version 8, Volumes 1 and 2](#)

Author Information

Tom Winn
 Texas State Auditor's Office
 P.O. Box 12067
 Austin, TX 78711-2067

Telephone: 512 / 936-9735
 E-mail: twinn@sao.state.tx.us