

Paper 307-2013

## **OUT= is on the way out. Use ODS OUTPUT instead**

Stanley Fogleman, Harvard Clinical Research Institute, Boston, MA

### **ABSTRACT**

There are, as a general rule, two methods to create a SAS® dataset from procedural output. The more traditional one is OUT=. This feature is being replaced by the ODS OUTPUT statement as new capabilities are added to procedures. In the future, only existing variables (generally pre 9.1) will be available on the OUT= statement. Therefore, it behooves the day-to-day SAS programmer to become familiar with the new syntax.

### **INTRODUCTION**

Anyone who has learned SAS in the last ten years knows that a good number of commonly used procedures have an OUT= parameter whose purpose is to specify a data set name. ODS, by contrast, was to provide a universal output destination. Just as DATA= is an input to many SAS procedures, OUT= is as close to a universal output parameter as one could find. The limitation in using the OUT= parameter is that it assumes you want all of the default values in your output dataset. Using ODS OUTPUT, by contrast, allows you to “pick and choose” what columns or subset of the output data would be most useful.

Originally, ODS was to provide a replacement for the former PRINTER based philosophy. Instead, ODS enabled output to many destinations: HTML, PRINTER and OUTPUT, among others. An ODS OUTPUT statement produces a SAS dataset from an output object and manages the selection and exclusion lists for the output procedure. The use of ODS OUTPUT requires a little more work, but it should allow a more granular approach to the output, allowing the user to select only what they truly need. Also, using the ODS OUTPUT statement, one can combine the output from several procedures and subset the data using BY groups or a combination of both.

### **WHERE ARE ODS AND OUT= THE SAME AND WHERE ARE THEY DIFFERENT?**

An OUT= statement on a procedure is self-evidently limited to the procedure in which it is called. OUT= produces whatever the original developers of the procedure felt were important. It is important to note that code which is working today with the OUT= parameter should continue to do so in the future.

An ODS OUTPUT statement produces a SAS dataset from an output object and manages the selection and exclusion lists for the output procedure. ODS produces output objects, which are particular to the procedure that produces them.

Here is an example of using SAS to produce a listing of output objects:

```
ods trace on /listing;  
  
proc univariate data=sashelp.class;  
  
var height weight;  
  
run;  
  
ods trace off;
```

OUT= is on the way out. Use ODS OUTPUT, instead, continued

```

Output Added:
-----
Name:      ExtremeObs
Label:     Extreme Observations
Template:  base.univariate.ExtObs
Path:     Univariate.Height.ExtremeObs
-----

```

| Extreme Observations |     |                |     |  |
|----------------------|-----|----------------|-----|--|
| ----Lowest----       |     | ----Highest--- |     |  |
| Value                | Obs | Value          | Obs |  |
| 51.3                 | 11  | 66.5           | 14  |  |
| 56.3                 | 13  | 66.5           | 19  |  |

**Output 1. Output from a ODS TRACE ON Statement**

In the LISTING window, look for the “Name:” under the heading “Output Added:”. This will be the name of the output object that you want. The five objects that should be listed are: Moments, Base Measures, Tests for Location, Quantiles, and Extreme Obs.

The next step is to select which items you are interested in (e.g. extremeobs), and a variable name to store them in (e.g. minmax). The general form of the ODS OUTPUT statement is: ODS OUTPUT output-object=data-set-name.<sup>1</sup>

For example:

```

ods listing close;

ods output extremeobs=minmax;

proc univariate data=sashelp.class;

var height weight;

run;
ods output close;
ods listing;

```

## SOME OTHER THINGS YOU CAN DO WITH ODS OUTPUT

It is possible to direct output from the same procedure to different datasets using ODS OUTPUT. It's difficult, but even though the physical act of printing is not involved in producing an output dataset, the mechanics are still very much in place. The desired ODS OUTPUT object will not be produced when NOPRINT is used.

Also, remember that an ODS OUTPUT object is only produced at a step boundary, so a "run" or "quit" statement is required.

Here are three important commands to remember when using ODS OUTPUT:

```
ods output clear;
ods output close;
ods output show;
```

The first sets the list for the OUTPUT destination to EXCLUDE ALL.

The second closes the output destination.

The third displays the current selection or exclusion list.

## HOW ABOUT PERSIST=PROC?

PERSIST=PROC is a method to produce a single output dataset from multiple procedures. Say you wanted to execute the same procedure more than once and concatenate the output.

For example:

```
ods listing close;

ods output extremeobs(PERSIST=PROC)=minmax;

proc univariate data=sashelp.class;

var height weight;

run;

proc univariate data=sashelp.shoes;

var color price;

run;
ods output close;
ods listing;
```

## WHAT IS MATCH\_ALL?

MATCH\_ALL is used for creating separate SAS datasets using BY/CLASS variables. This is significantly easier than creating and subsetting data which usually involves multiple datasets.

For example:

```
ods listing close;

ods output extremeobs(MATCH_ALL)=sample;

proc univariate data=sashelp.class;

var height weight;

run;
```

OUT= is on the way out. Use ODS OUTPUT, instead, continued

## CONCLUSION

Mastering the ODS OUTPUT statement has many rewards for the SAS programmer, not the least of which is the ability to combine and subset data from multiple procedures without extensive extra programming.

## REFERENCES

Haworth, Lauren, Zender, Cynthia, Burlew, Michele. 2009. Output Delivery System : The Basics and Beyond. pp.211-231. Cary, North Carolina: SAS Press

Carpenter, Art. 2012. Carpenter's Guide to Innovative SAS Techniques. pp 326-332. Cary, North Carolina: SAS Press.

SAS Institute. 2011. SAS 9.3 Output Delivery System User's Guide. p. 448. Cary, North Carolina: SAS Press

## ACKNOWLEDGMENTS

To Cynthia Zender, for her advice and encouragement in the writing of this paper.

Chris Abbate for reviewing the paper and making suggestions.

## CONTACT INFORMATION

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

Name: Stanley Fogleman  
Enterprise: Harvard Clinical Research Institute  
Address: 930 Commonwealth Ave West, 3<sup>rd</sup> Floor  
City, State ZIP: Boston, MA 02215  
E-mail: stanleydotfoglemanathcridotharvarddotedu

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.