Paper 388-2014

Sailing Over the ACROSS Hurdle in PROC REPORT

Cynthia L. Zender, SAS Institute Inc., Cary, NC

ABSTRACT

To get the full benefit from PROC REPORT, the savvy programmer needs to master ACROSS usage and the COMPUTE block. Timing issues with PROC REPORT and absolute column references can unlock the power of PROC REPORT.

This presentation illustrates how to make the most of ACROSS usage with PROC REPORT. Use PROC REPORT instead of multiple TRANSPOSE steps. Find out how to use character variables under an across usage item. Learn how to impact the column headers for ACROSS items. Learn how to use aliases. Find out how to perform row-wise trafficlighting and trafficlighting based on multiple conditions.

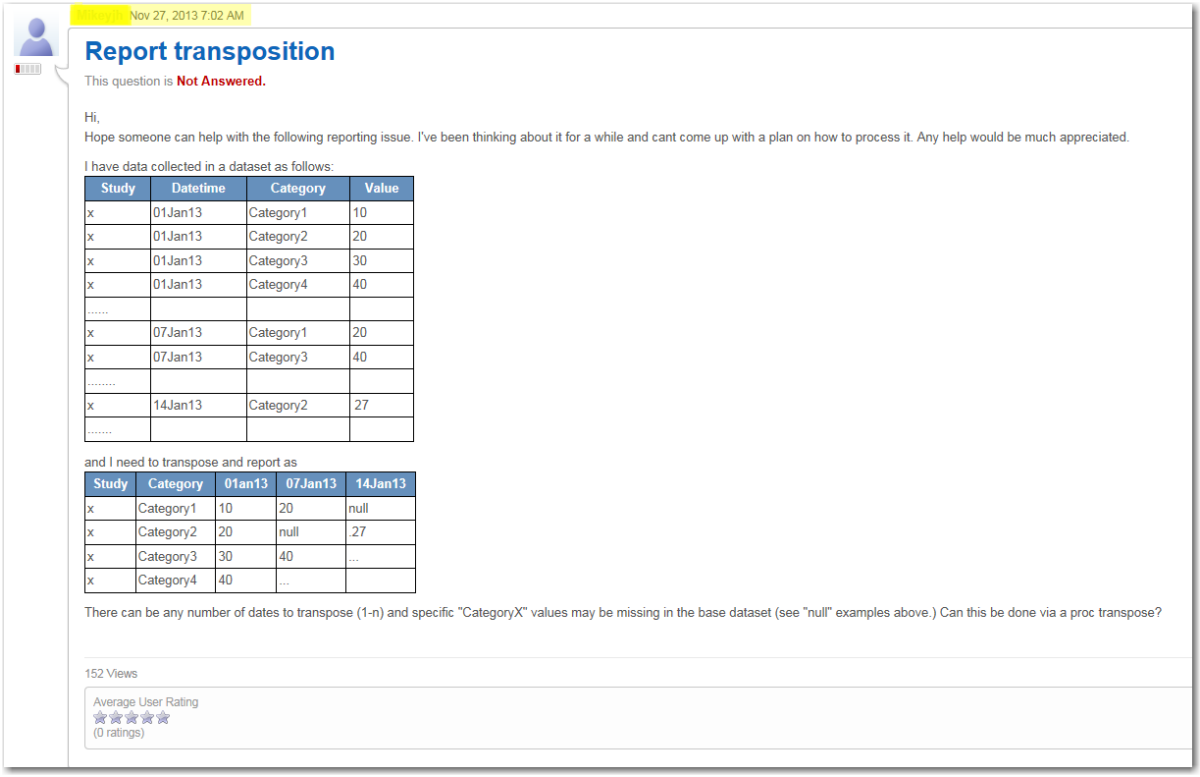
INTRODUCTION

At first glance, using ACROSS with PROC REPORT seems like a walk in the park. You like PROC REPORT syntax better than PROC TABULATE syntax. You like not having the big empty box area at the intersection of row headers and column headers. You like the PROC REPORT spanning headers. But then, all of a sudden, you need to use a character variable as the ACROSS item on the report.

Or, perhaps you just don't know about PROC REPORT and all it can do. So, I'm going to start the paper with a simple usage of ACROSS.

WITH A LITTLE HELP FROM MY FRIENDS

Display 1 shows a SAS Community forum question (or part of it). To review the full posting and all the responses, visit the Community forum website (**`https://communities.sas.com/message/189755#189755`**).



Nov 27, 2013 7:02 AM

Report transposition

This question is **Not Answered.**

Hi,

Hope someone can help with the following reporting issue. I've been thinking about it for a while and cant come up with a plan on how to process it. Any help would be much appreciated.

I have data collected in a dataset as follows:

Study	Datetime	Category	Value
x	01Jan13	Category1	10
x	01Jan13	Category2	20
x	01Jan13	Category3	30
x	01Jan13	Category4	40
.....			
x	07Jan13	Category1	20
x	07Jan13	Category3	40
.....			
x	14Jan13	Category2	27
.....			

and I need to transpose and report as

Study	Category	01an13	07Jan13	14Jan13
x	Category1	10	20	null
x	Category2	20	null	27
x	Category3	30	40	...
x	Category4	40	...	

There can be any number of dates to transpose (1-n) and specific "CategoryX" values may be missing in the base dataset (see "null" examples above.) Can this be done via a proc transpose?

152 Views

Average User Rating

(0 ratings)

Display 1. SAS Community Forum Post with a "TRANSPOSE" Question

The original poster (OP) of this question got quite a few responses that jumped immediately on his PROC TRANSPOSE question. But, one crucial piece of the question said that the OP needed to transpose and "report as". PROC TRANSPOSE is a great procedure, but it creates a SAS data set, not a report. So, if the OP did a transpose and then the ARRAY processing that might be required after the transposition, that would be two passes through the data. And that's just two passes if you don't count the PROC PRINT for the report piece.

Both PROC TABULATE and PROC REPORT would have produced a "transposed" report with every unique date value as a column and the sum of the VALUE variable inside the data cells. Assuming the data looks like the original data set example posted in the track, consider the code below. It produced the output shown in Output 1.

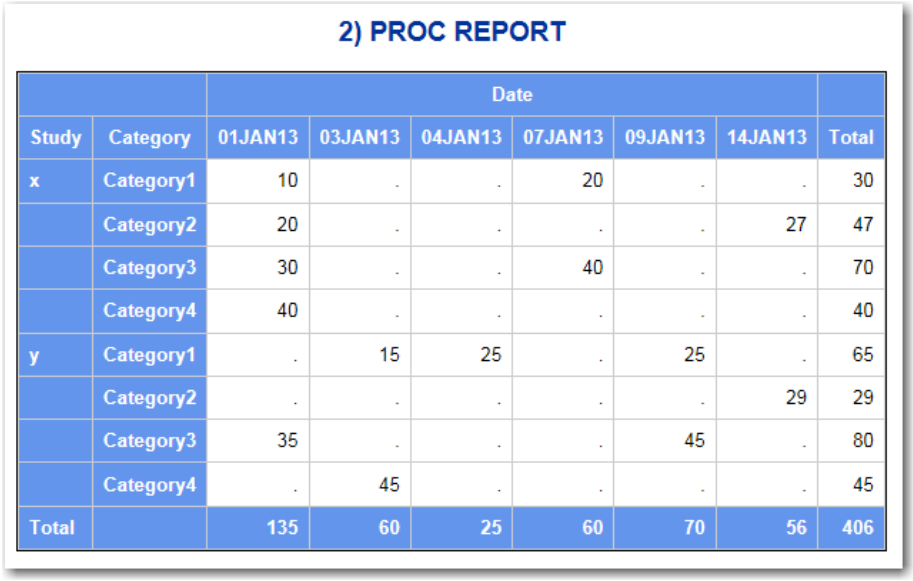
```
ods html file='c:\temp\want_report.html' style=sasweb;

proc tabulate data=ex1 f=comma6.;
  title '1) PROC TABULATE';
  class study datetime category;
  var value;
  table study*Category all*{style=Header},
        datetime='Date'*value=' ' all*value=' ';
  keylabel sum=' '
           all='Total';
  keyword all / style={vjust=b};
  format datetime date7.;
run;

proc report data=ex1 nowd
  style(summary)=Header;
  title '2) PROC REPORT';
  column study category value,datetime value=tot;
  define study / group style(column)=Header;
  define category / group style(column)=Header;
  define datetime /across order=internal f=date7. 'Date';
  define value / analysis sum f=comma6. ' ';
  define tot / analysis sum 'Total' f=comma6. ;
  rbreak after / summarize;
  compute after;
    study='Total';
  endcomp;
run;

ods html close;;
```

The PROC REPORT and TABULATE output is the same, so only the REPORT results are shown in Output 1. In addition, both PROC REPORT and PROC TABULATE can produce grand totals for the report or as a final column. Notice that PROC REPORT uses the DATETIME variable as an ACROSS item on the report definition. Also notice that, in the COLUMN statement, the syntax that puts VALUE underneath (or nested within) each DATETIME is the use of the comma operator:

```
column study category value,datetime value=tot;
```

The comma operator in this instance is causing VALUE to be summarized underneath every unique value for DATETIME. Then, VALUE is used a second time on the report to produce the final TOTAL column.

So, PROC TRANSPOSE might have been the first choice for the OP, but by using one of the "powerhouse" report procedures capable of creating a cross-tabular report, the final result was produced without using PROC TRANSPOSE.



2) PROC REPORT

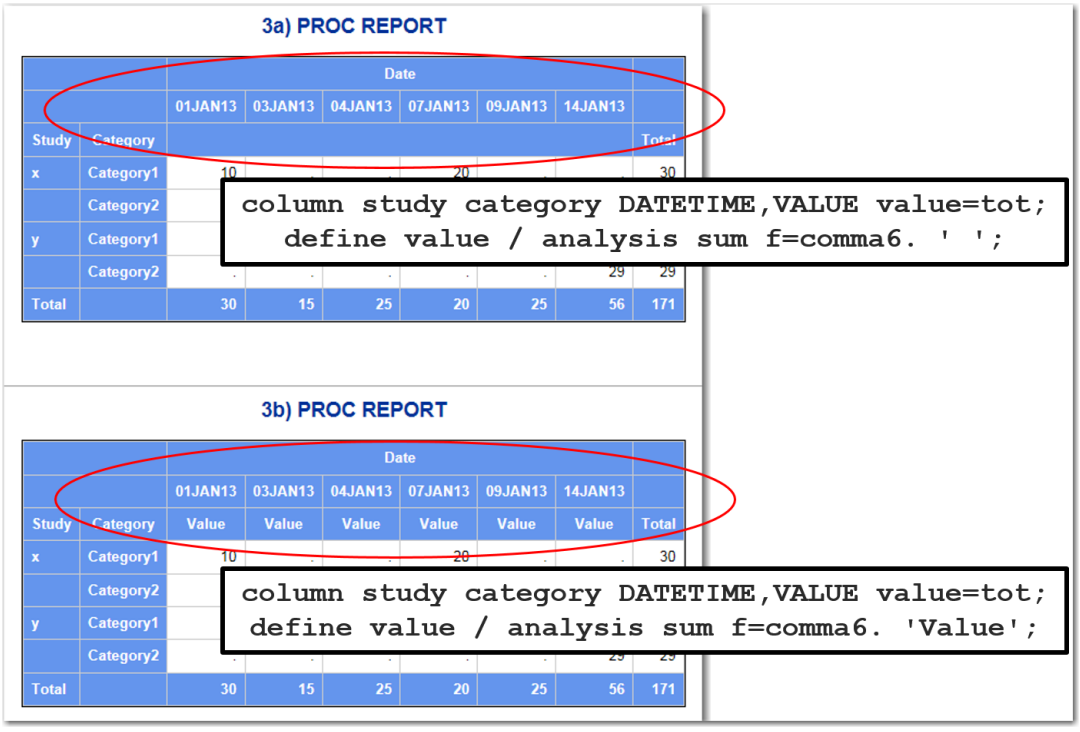
		Date						
Study	Category	01JAN13	03JAN13	04JAN13	07JAN13	09JAN13	14JAN13	Total
x	Category1	10	.	.	20	.	.	30
	Category2	20	.	.	.	.	27	47
	Category3	30	.	.	40	.	.	70
	Category4	40	.	.	.	.	.	40
y	Category1	.	15	25	.	25	.	65
	Category2	.	.	.	.	.	29	29
	Category3	35	.	.	.	45	.	80
	Category4	.	45	.	.	.	.	45
Total		135	60	25	60	70	56	406

Output 1. Output from PROC REPORT with Transposed Columns and Totals

Let's take a brief look at one of the hurdles that people frequently face when they first use the comma operator with PROC REPORT. What if the order of VALUE and DATETIME had been reversed in the COLUMN statement?

```
column study category datetime,value value=tot;
```

Then the results would take show that PROC REPORT has allocated space for the header, even when the header is blanked out, as shown in Output 2.



Output 2. Output from PROC REPORT with a Different COLUMN Statement

There is a way to overcome this automatic allocation of space for the column header. The method is to use spanning headers in the column statement and make sure that the whole row (where the repeated headers for "Value" appear) is blank. PROC REPORT will suppress a row that is entirely composed of blanks. This final report is shown in Output 3, using the code below. The ODS HTML statements are not shown for this code snippet.

```
proc report data=ex1 nowd
  style(summary)=Header;
  where category in ('Category1', 'Category2');
  title '3c) PROC REPORT';
  column ('Study' study) ('Category' category) datetime,value ('Total' value=tot);
  define study / group style(column)=Header ' ';
  define category / group style(column)=Header ' ';
  define datetime /across order=internal f=date7. 'Date';
  define value / analysis sum f=comma6. ' ';
  define tot / analysis sum ' ' f=comma6. ;
  rbreak after / summarize;
  compute after;
    study='Total';
  endcomp;
run;
```

3c) PROC REPORT

		Date						
Study	Category	01JAN13	03JAN13	04JAN13	07JAN13	09JAN13	14JAN13	Total
x	Category1	10	.	.	20	.	.	30
	Category2	20	.	.	.	.	27	47
y	Category1	.	15	25	.	25	.	65
	Category2	.	.	.	.	.	29	29
Total		30	15	25	20	25	56	171

Output 3. Output from PROC REPORT with Spanning Headers

Notice that the COLUMN statement still shows **`DATETIME,VALUE`** but most of the other variables have their headers specified in the COLUMN statement, which gives PROC REPORT an entirely blank header row, which it can suppress.

So, either way the ACROSS items are defined and given headers, and it is possible to get PROC REPORT to transpose and summarize data for report purposes without creating an output data set first.

PLEASE, LOUISE PULL ME OFF A MY KNEES

Sometimes though, you don't want PROC REPORT to summarize data in the data cells; sometimes you want to display values underneath columns and the variables are character variables. In this case, PROC REPORT can bring you to your knees. What if you have some team data that shows the captain for every combination of League and Team. And, what you want is a transposed report that shows a League on every row and a Team for every column, with the Captain value (a character variable) in the data cell. That is exactly what another SAS Community forum question (or part of it) asked. To review the full posting and all the responses, visit the SAS Community forum (**`https://communities.sas.com/message/168440#168440`**). Here's what the revised sample data looks like. All the variables are character variables in the WORK.EX2 data set.

	League	Team	Captain
1	N	A	Alan
2	N	B	Bob
3	N	C	Charles
4	S	A	David
5	S	B	Edward
6	S	C	Frank

Display 2. Data Set with Character Variables

As described, the report should show the captain's name in every data cell, as shown in Output 4.

Final Report

	Team		
League	A	B	C
N	Alan	Bob	Charles
S	David	Edward	Frank

Output 4. Desired Results with ACROSS and Character Variables

Again, PROC REPORT helps you jump this hurdle. But not without some stumbles. The first attempt, shown in the screenshot of the SAS Log below, tells you what the problem is. There is no statistic associated with the variable CAPTAIN (which is nested underneath the ACROSS variable, TEAM).

```
889
890  proc report data=ex2 nowd;
891    title 'lst try';
892    column league team,captain;
893    define league / group;
894    define team / across;
895    define captain / display;
896  run;

ERROR: There is no statistic associated with Captain.
NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE REPORT used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

Display 3. SAS Log After First Try

There are two ways to generate the desired report. One way is to put a "dummy" numeric variable on the report, and the other way is to put a calculated statistic (such as N) on the report. The code below produces Output 5.

```
proc report data=ex2 nowd;
  title1 '2nd try with dummyvar';
  column league team,captain dummyvar;
  define league / group;
  define team / across;
  define captain / display;
  define dummyvar / computed;
  compute dummyvar;
    dummyvar = 1;
  endcomp;
run;
```

Note how the DUMMYVAR computed item is assigned a value of 1 in the COMPUTE block. Eventually, this item will be defined as NOPRINT, which means that the number being assigned doesn't make a difference.

2nd try with dummyvar

	Team			
	A	B	C	
League	Captain	Captain	Captain	dummyvar
N	Alan	Bob	Charles	1
S	David	Edward	Frank	1

Output 5. Creating a COMPUTED item called DUMMYVAR

However, note the difference in the value when the N statistic is requested in the code. The code below produces Output 6.

```
proc report data=ex2 nowd;
  title1 'Use N instead of DUMMYVAR';
  column league team,captain n;
  define league / group;
  define team / across;
  define captain / display;
  define n / 'Count';
run;
```

If you compare the N or count value for each row, you will see it is 3 versus the 1 from the other report.

Use N instead of DUMMYVAR

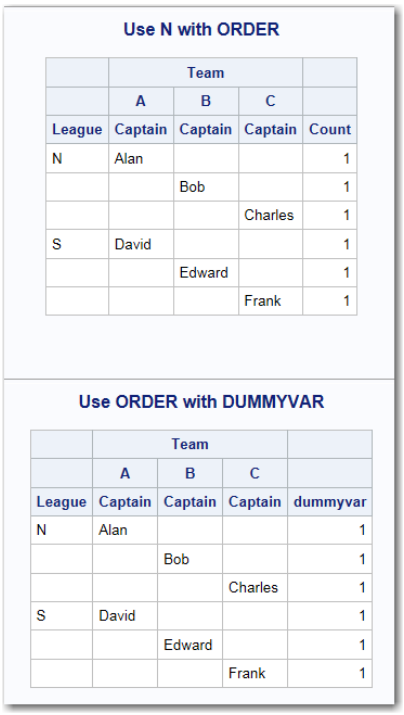
	Team			
	A	B	C	
League	Captain	Captain	Captain	Count
N	Alan	Bob	Charles	3
S	David	Edward	Frank	3

Output 6. Using the N Statistic

Using GROUP with ACROSS and having a numeric variable on the report is what allows this “transposed” report to use character variable values in the data cells and display only one row per League value. If you did not use GROUP for the first column, but used ORDER instead, then you would see what I call “stairstep” output, as shown in Output 7.

This will be true whether you use the “dummyvar” technique or the “N” technique. The code snippet below (showing the usage of League as ORDER) is the only change made to either of the previous two programs.

```
define league / order;
```



Use N with ORDER

	Team			
	A	B	C	
League	Captain	Captain	Captain	Count
N	Alan			1
		Bob		1
			Charles	1
S	David			1
		Edward		1
			Frank	1

Use ORDER with DUMMYVAR

	Team			
	A	B	C	
League	Captain	Captain	Captain	dummyvar
N	Alan			1
		Bob		1
			Charles	1
S	David			1
		Edward		1
			Frank	1

Output 7. "Stairstep" Output using ORDER and ACROSS with Character VARIABLE

The only change to the original program to produce the final report, as shown in Output 4, was putting the NOPRINT option on either the DEFINE statement for DUMMYVAR

```
define dummyvar / computed noprint;
```

or the DEFINE statement for N.

```
define n / 'Count' noprint;
```

So far, using ACROSS has allowed us to take data in one form and create a cross-tabular report without using PROC TRANSPOSE; without making multiple passes through the data; and without using PROC TABULATE. And it's allowed us to do something unique, such as showing the value of a character variable instead of a numeric variable underneath the unique values of the ACROSS variable. Next, we're going to tackle absolute column numbers and PROC REPORT.

OFF WE GO, INTO THE WILD BLUE YONDER

We will launch ourselves so high over the absolute column number hurdle that we're going to be touching the clouds in the blue sky! So far, the ACROSS items we've used haven't been that difficult, mostly because we haven't needed to calculate a new item nested under an ACROSS variable.

All that's going to change. Consider this code:

```
proc report data=sashelp.class nowd out=abscols
  style(summary)=Header;
  where age le 13;
  title '1) Proc Report Crosstab Report';
  column age ( sex,(weight height))
             ('Overall' weight=wta height=hta);
  define age / group style(column)=Header;
  define sex / across 'Gender Avg';
  define weight / mean f=7.2;
  define height / mean f=7.2;
  define wta / mean f=7.2 style(column)=Header;
  define hta / mean f=7.2 style(column)=Header;
  rbreak after / summarize;
```

```
run;
```

The results of this program are shown in Output 8.

1) Proc Report Crosstab Report

	Gender Avg					
	F		M		Overall	
Age	Weight	Height	Weight	Height	Weight	Height
11	50.50	51.30	85.00	57.50	67.75	54.40
12	80.75	58.05	103.50	60.37	94.40	59.44
13	91.00	60.90	84.00	62.50	88.67	61.43
	78.80	57.84	95.90	60.22	87.35	59.03

Output 8. Two Variables under Each ACROSS Variable Value

So far, so good. If you've used PROC REPORT before, the COLUMN statement is straightforward. The AGE variable is a GROUP usage; while the SEX, the ACROSS variable, is nested with WEIGHT and HEIGHT. This is how a separate set of columns appears in Output 8 for the F value and another separate set of columns appears for the M value. Then, WEIGHT and HEIGHT are assigned aliases (WTA and HTA, respectively) and used a second time on the report to give the overall mean or average for every row.

But, what if you need to calculate the DIFFERENCE between WEIGHT and HEIGHT for Females and WEIGHT and HEIGHT for Males? Oh, and also calculate the DIFFERENCE between the OVERALL WEIGHT and HEIGHT for the AGE value?

The first syntax you might try is shown below.

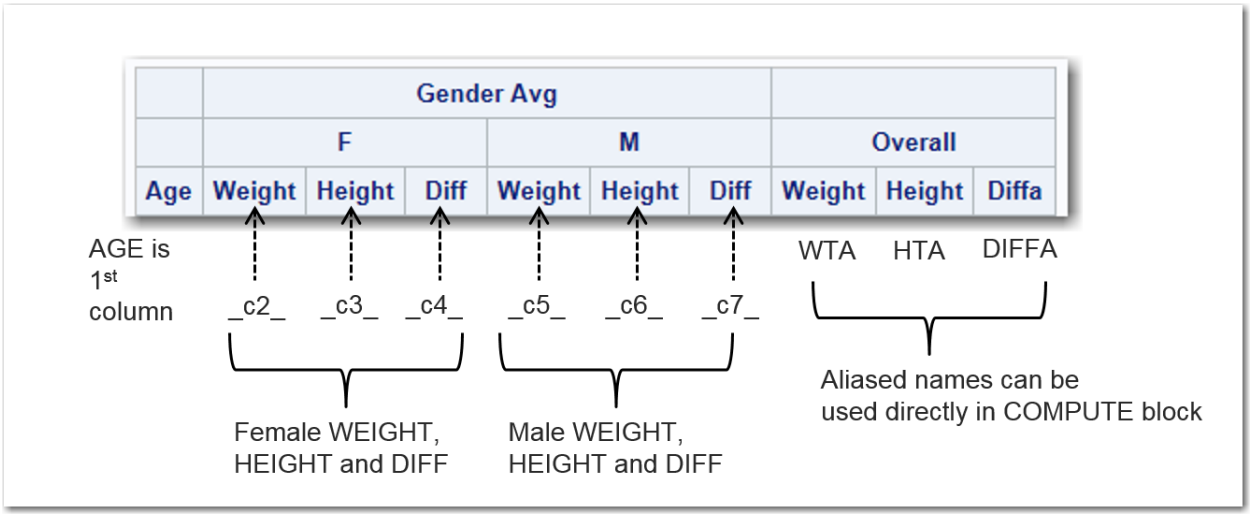
```
proc report data=sashelp.class nowd
  style(summary)=Header;
  where age le 13;
  title '2a) Proc Report with Calculated Column Specified Incorrectly';
  column age ( sex,(weight height Diff))
             ('Overall' weight=wta height=hta Diffa);
  define age / group style(column)=Header;
  define sex / across 'Gender Avg';
  define weight / mean f=7.2;
  define height / mean f=7.2;
  define diff / computed f=7.2;
  define wta / mean f=7.2 style(column)=Header;
  define hta / mean f=7.2 style(column)=Header;
  define diffa / computed f=7.2 style(column)=Header;
  compute diff;
     diff = weight.mean - height.mean ;
  endcomp;
  compute diffa;
     diffa = wta - hta;
  endcomp;
  rbreak after / summarize;
run;
```

You were a good PROC REPORT programmer. You used WEIGHT.MEAN and HEIGHT.MEAN in your COMPUTE block. But, it's not going to work. The SAS Log for the 2a report is shown in Display 4. Ah, now, you're singing the blues. PROC REPORT treats you so mean!

```
ERROR: The variable type of WEIGHT.MEAN is invalid in this context.
ERROR: The variable type of HEIGHT.MEAN is invalid in this context.
NOTE: The preceding messages refer to the COMPUTE block for Diff.
NOTE: Will not run due to compilation errors.
NOTE: The SAS System stopped processing this step because of errors.
NOTE: There were 10 observations read from the data set SASHELP.CLASS.
      WHERE age<=13;
NOTE: PROCEDURE REPORT used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds
```

Display 4. SAS Log with Incorrect Usage for ACROSS Nested Items

But, there were no complaints about the COMPUTE block for DIFFA! Why does PROC REPORT do that? When you use an ACROSS item on your report, PROC REPORT has a pre-processing step that it performs before it ever starts writing report rows. Behind the scenes, PROC REPORT assigns absolute column numbers to every item under the ACROSS item. So let's take a look at what that really means for the original report, with the DIFF and DIFFA columns added to the headers.



Display 5. COLUMN Headers Showing Absolute Column Numbers Assigned by PROC REPORT

Now, the formula that we need is easy to see. The Female difference needs to be calculated as:

Female Diff Value = Female Weight – Female Height or

_c4_ = _c2_ - _c3_

and the Male difference column needs to be calculated as:

Male Diff Value = Male Weight – Male Height or

_c7_ = _c5_ - _c6_.

And this is the only change to the COMPUTE block that we need to make to get the desired differences calculated.

```
compute diff;
    _c4_ = _c2_ - _c3_;
    _c7_ = _c5_ - _c6_;
endcomp;
```

The output from this change to the program is shown in Output 9.

2b) Proc Report with Calculated Column Specified Correctly

	Gender Avg								
	F			M			Overall		
Age	Weight	Height	Diff	Weight	Height	Diff	Weight	Height	Diffa
11	50.50	51.30	-0.80	85.00	57.50	27.50	67.75	54.40	13.35
12	80.75	58.05	22.70	103.50	60.37	43.13	94.40	59.44	34.96
13	91.00	60.90	30.10	84.00	62.50	21.50	88.67	61.43	27.23
	78.80	57.84	20.96	95.90	60.22	35.68	87.35	59.03	28.32

Output 9. Difference Calculated Correctly

And, now, I know you're thinking, "But wait! What if I have more than two ACROSS values? What if I don't know how many ACROSS values I might have from report period to report period?" There's a macro program for that. Look in Allison McMahill's 2007 paper on PROC REPORT.

What happens if I not only have a calculated item, but I want to do trafficlighting on that item and other items related to it? Take a look at Output 10 to see what I mean.

		Product								
		BED			CHAIR			DESK		
Country	Division	Actual Sales	Predicted Sales	Pct of Predict	Actual Sales	Predicted Sales	Pct of Predict	Actual Sales	Predicted Sales	Pct of Predict
CANADA	CONSUMER	$24,176.00	$21,197.00	114.05%	$22,493.00	$23,025.00	97.69%	$26,234.00	$25,329.00	103.57%
	EDUCATION	$23,553.00	$23,018.00	102.32%	$27,746.00	$23,771.00	116.72%	$25,953.00	$24,064.00	107.85%

Output 10. Using ACROSS Values to Perform Trafficlighting

In the above report snippet, the columns for Actual Sales and Predicted Sales under each Product value are given a background color to match the background color for the Pct of Predict column. So more absolute columns are going to be involved. There is a user-defined format to create the background color for the percent column.

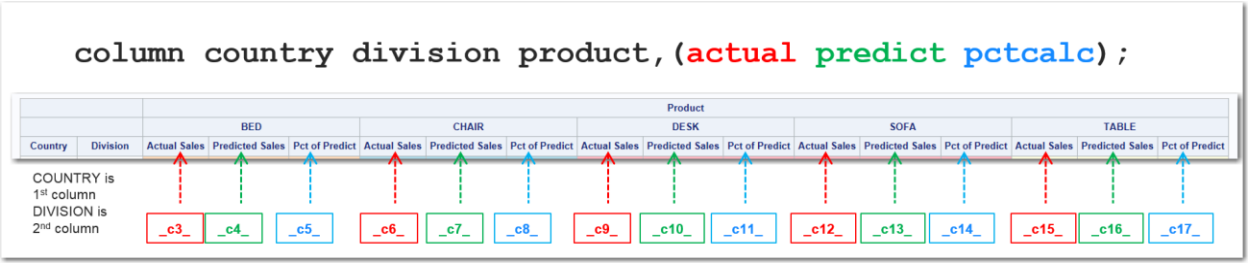
```
proc format;
  value fpct low-<.95='lightyellow'
              .95-<1.00 = 'lightblue'
             1.00-<1.075 = 'lightpink'
             1.075-<1.10='lavender'
             1.10-<1.50 = 'peachpuff'
             1.50-high='verylightgreen';
run;
```

In this COMPUTE block, all the values were hard-coded for the absolute column numbers. This program has not been "macroized", so it uses the hard-coded values for the absolute column numbers. Basically, what you need to know is that Actual Sales for the BED product is _c3_, Predicted Sales for the BED product is _c4_, and the calculated Pct of Predict for the BED product is _c5_.

This means that there is a "set" of three columns underneath every value for the PRODUCT variable. So, Actual Sales for the DESK product will be _c6_, for Predicted sales _c7_, and for DESK Pct of Predict the absolute column number will be _c8_. Study Display 6 to see the absolute column numbers for every column underneath every product. The COLUMN statement is shown in the screenshot. The DEFINE usage for COUNTRY and DIVISION is GROUP. The DEFINE usage for PRODUCT is ACROSS. The DEFINE statement for PRTCALC specifies the user-defined format in a STYLE= override:

```
define pctcalc / computed 'Pct of Predict' f=percent9.2
                 style(column)={background=fpct.};
```

This is the simplest form of trafficlighting to use with PROC REPORT. The secondary form of trafficlighting that you can perform with PROC REPORT involves using the CALL DEFINE statement. This is what must be used for the Actual and Predicted Sales columns underneath each product. But, again, I need to know that there are five products and that will help me verify the absolute column numbers for my code (or my macro program).



Display 6. COLUMN Headers Showing COLUMN statement and Absolute Column Numbers

One of the things that makes this report tricky is that I need to know the color for _C5_, _c8_, and so on, before I can set the background color for Actual and Predicted Sales columns. Luckily, PROC REPORT helps us out. Although I cannot specify the CALL DEFINE for Actual and Predicted Sales in a separate COMPUTE block, PROC REPORT allows me to put a CALL DEFINE statement in the Pct of Predict COMPUTE block (the computed item is called PCTCALC) for any variable that is to the left of the PCTCALC column. This makes it possible for me to calculate each of my percent values in one COMPUTE block and to assign colors in the same COMPUTE block.

When PROC REPORT executes the COMPUTE block, it knows which data cells have already been put on the report row and it uses the correct assignment statement.

Next, I need a very long text string as my style variable value for the CALL DEFINE statement. Note that I have a LENGTH statement in my COMPUTE block for the temporary variable SVAR5 (and there are more), one for each PCTCALC column. Again, these variables are hard-coded in the program. If I wanted to “macroize” the COMPUTE block, this would be a good section of code to turn into a macro call. Finally, I use the SVAR5 variable in the CALL DEFINE statements for _c3_ (Actual Sales for BED product) and for _c4_ (Predicted Sales for BED product).

```
  compute pctcalc;
    length svar5 svar8 svar11 svar14 svar17  $50;
    _c5_ = _c3_ / _c4_;
    _c8_ = _c6_ / _c7_;
    _c11_ = _c9_ / _c10_;
    _c14_ = _c12_ / _c13_;
    _c17_ = _c15_ / _c16_;
    svar5 = 'style={background='||put(_c5_,fpct.)||'}';
. . . more svar creation . . .
    call define('_c3_','style',svar5);
    call define('_c4_','style',svar5);
. . . more call define statements. . .
  endcomp;
```

This is the abbreviated version of the COMPUTE block. To see the full code for the COMPUTE block, download the ZIP file of programs that will out on the R&D website on support.sas.com. As a bonus, there is a “macroized” version of this program in the ZIP file, too.

In our next example, I’m going to use ACROSS to change the background color of a spanning header. Consider this code and the simplified report output that is produced, as shown in Output 11. There are no absolute columns in this code because we are focusing on the spanning headers.

```
proc report data=sashelp.prdsale nowd;
  column ('Group Variables' country division) product,(actual predict);
  define country / group style(header)={background=lightgreen};
  define division / group style(header)={background=lightgreen};
  define product / across ' ' style(header)={background=lightyellow};
  define actual / sum style(header)={background=lightyellow};
  define predict / sum style(header)={background=lightyellow};
run;
```

Group Variables		BED		CHAIR		DESK		SOFA		TABLE	
Country	Division	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales
CANADA	CONSUMER	$24,176.00	$21,197.00	$22,493.00	$23,025.00	$26,234.00	$25,329.00	$24,803.00	$23,374.00	$20,278.00	$25,394.00
	EDUCATION	$23,553.00	$23,018.00	$27,746.00	$23,771.00	$25,953.00	$24,064.00	$25,332.00	$22,352.00	$26,422.00	$21,495.00
GERMANY	CONSUMER	$24,294.00	$23,905.00	$21,217.00	$20,282.00	$24,608.00	$22,740.00	$28,690.00	$26,435.00	$25,037.00	$23,081.00
	EDUCATION	$21,840.00	$19,891.00	$25,888.00	$23,787.00	$23,894.00	$21,899.00	$26,370.00	$23,082.00	$24,160.00	$26,452.00
U.S.A.	CONSUMER	$25,230.00	$25,089.00	$26,304.00	$23,091.00	$23,430.00	$27,745.00	$21,586.00	$23,811.00	$24,105.00	$23,087.00
	EDUCATION	$22,944.00	$24,767.00	$24,632.00	$22,154.00	$25,113.00	$24,418.00	$21,807.00	$21,397.00	$22,198.00	$26,163.00

Output 11. Spanning Header Is Original Background Color

There is one easy way to fix this, if what you want is both Country and Division to be light green and to have them share a spanning header. That method is to specify a style override for the header color in the PROC REPORT statement and then alter the PRODUCT and numeric variable headers.

```
proc report data=sashelp.prdsale nowd
  style(header)={background=lightgreen};
  column ('Group Variables' country division) product,(actual predict);
```

The full code which produced Output 12 is contained in the ZIP file of programs for the paper.

Group Variables		BED		CHAIR		DESK		SOFA		TABLE	
Country	Division	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales
CANADA	CONSUMER	$24,176.00	$21,197.00	$22,493.00	$23,025.00	$26,234.00	$25,329.00	$24,803.00	$23,374.00	$20,278.00	$25,394.00

Output 12. Easy Background for Single Spanning Headers

As shown in Output 13, it is also the easy code method that allows COUNTRY and DIVISION to both have a light green background color for separate spanning headers.

Three Values	Current	BED		CHAIR		DESK		SOFA		TABLE	
Country	Division	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales
CANADA	CONSUMER	$24,176.00	$21,197.00	$22,493.00	$23,025.00	$26,234.00	$25,329.00	$24,803.00	$23,374.00	$20,278.00	$25,394.00

Output 13. Easy Background for Multiple Spanning Headers

But what if you wanted multiple spanning headers and you wanted the Country headers to be light green and Division headers to be the “peachpuff” color, as shown in Output 14?

Three Values	Current	BED		CHAIR		DESK		SOFA		TABLE	
Country	Division	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales	Actual Sales	Predicted Sales
CANADA	CONSUMER	$24,176.00	$21,197.00	$22,493.00	$23,025.00	$26,234.00	$25,329.00	$24,803.00	$23,374.00	$20,278.00	$25,394.00

Output 14. Different Colors for Spanning Headers

This is when we turn to ACROSS variables to help out. First, we need to add some helper variables to a data set. Here’s the complete code to make the new version of the SASHELP.PRDSALE data set. The helper variables are FAKEVAR, FAKEDIV, X, and Y. FAKEVAR is assigned the same value as COUNTRY; and FAKEDIV is assigned the same value as DIVISION. X is set to the string “Three Values,” which will span COUNTRY, and Y is set to the string “Current,” which will span DIVISION.

```
data prdsale;
  length x y fakevar fakediv $25;
  set sashelp.prdsale;
  x='Three Values';
  y='Current';
  fakevar = country;
  fakediv = division;
run;
```

Let’s look at a partial display of the WORK.PRDSALE data set.

	x	y	fakevar	fakediv	Actual Sales	Predicted Sales	Country	Region	Division	Product type	Product
1	Three Values	Current	CANADA	CONSUMER	$5.00	$425.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
2	Three Values	Current	CANADA	CONSUMER	$164.00	$215.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
3	Three Values	Current	CANADA	CONSUMER	$422.00	$948.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
4	Three Values	Current	CANADA	CONSUMER	$424.00	$544.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
5	Three Values	Current	CANADA	CONSUMER	$854.00	$764.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
6	Three Values	Current	CANADA	CONSUMER	$168.00	$446.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
7	Three Values	Current	CANADA	CONSUMER	$8.00	$957.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
8	Three Values	Current	CANADA	CONSUMER	$748.00	$967.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
9	Three Values	Current	CANADA	CONSUMER	$682.00	$11.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
10	Three Values	Current	CANADA	CONSUMER	$300.00	$110.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
11	Three Values	Current	CANADA	CONSUMER	$672.00	$263.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
12	Three Values	Current	CANADA	CONSUMER	$894.00	$215.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA
13	Three Values	Current	CANADA	CONSUMER	$284.00	$414.00	CANADA	EAST	CONSUMER	FURNITURE	BED
14	Three Values	Current	CANADA	CONSUMER	$705.00	$770.00	CANADA	EAST	CONSUMER	FURNITURE	BED

Display 7. Partial Viewtable Display of WORK.PRDSALE

Now, using WORK.PRDSALE for PROC REPORT, we use FAKEVAR and FAKEDIV to guarantee that the rows are going to be in the same order as on the original report. That's needed because now X is going to become an ACROSS item that is crossed with COUNTRY using the comma operator, and Y is an ACROSS item that is crossed with DIVISION.

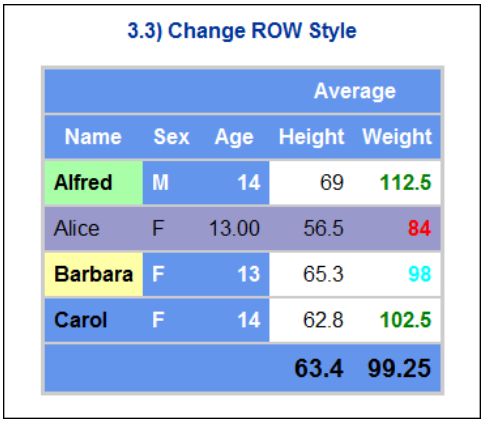
```
proc report data=prdsale nowd out=abscol;
  column fakevar fakediv x,(country) y,(division) product,(actual predict);
  define fakevar / group noprint ' ';
  define fakediv / group noprint ' ';
  define x / across ' ' style(header)={background=lightgreen};
  define country / group style(header)={background=lightgreen};
  define y / across ' ' style(header)={background=peachpuff};
  define division / group style(header)={background=peachpuff};
  define product / across ' ' style(header)={background=lightyellow};
  define actual / sum style(header)={background=lightyellow};
  define predict / sum style(header)={background=lightyellow};
run;
```

Since PROC REPORT requires that there is a GROUP or ORDER item to the left of an ACROSS item, FAKEVAR and FAKEDIV will be used to maintain the order of the rows. They do not need to be displayed on the report, which is why they are specified as NOPRINT items in the DEFINE statement.

With X as the ACROSS item nested on top of COUNTRY, both X and COUNTRY can have their background color changed to light green. With Y as the ACROSS item nested on top of DIVISION, both Y and DIVISION can have their background color changed to peachpuff, as shown in Output 14, based on the above code.

In our last example, I want to return to the concept of trafficlighting in order to showcase something that PROC REPORT does well, and which cannot be done by other procedures. PROC REPORT can perform trafficlighting based on multiple conditions. This is something that I discussed in my paper last year, with this example from SASHELP.CLASS:

```
   compute age;
    if age = 13 and name = 'Alice' then do;
       call define(_row_,'style','style={background=cx9999cc}');
       call define(_col_,'format','5.2');
    end;
  endcomp;
```



3.3) Change ROW Style

		Average		
Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13.00	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
			63.4	99.25

Output 15. Trafficlighting Based on Multiple Conditions

But, last year's example did NOT use ACROSS variables. Let's revisit this earlier report:

		Product								
		BED			CHAIR			DESK		
Country	Division	Actual Sales	Predicted Sales	Pct of Predict	Actual Sales	Predicted Sales	Pct of Predict	Actual Sales	Predicted Sales	Pct of Predict
CANADA	CONSUMER	$24,176.00	$21,197.00	114.05%	$22,493.00	$23,025.00	97.69%	$26,234.00	$25,329.00	103.57%
	EDUCATION	$23,553.00	$23,018.00	102.32%	$27,746.00	$23,771.00	116.72%	$25,953.00	$24,064.00	107.85%

Output 16. Using ACROSS Values to Perform Trafficlighting

What if you wanted trafficlighting to be performed only on the CHAIR and DESK columns when COUNTRY was CANADA? This means trafficlighting needs to be performed based on a GROUP item value and on an ACROSS item value.

Because we now understand how absolute column numbers work, we understand that the CHAIR items are columns _c6_, _c7_, and _c8_, while the DESK items are columns _c9_, _c10_, and _c11_. This makes it easy to perform trafficlighting for the CANADA cells based on those two pieces of information. In this last example, the user-defined format was changed slightly to highlight only the cells where the value was GT 100% for the Pct of Predict column for CHAIR and DESK in Canada. Partial results are shown in Output 17.

		Product														
		BED			CHAIR			DESK			SOFA			TABLE		
Country	Division	Actual Sales	Predicted Sales	Pct of Predict	Actual Sales	Predicted Sales	Pct of Predict	Actual Sales	Predicted Sales	Pct of Predict	Actual Sales	Predicted Sales	Pct of Predict	Actual Sales	Predicted Sales	Pct of Predict
CANADA	CONSUMER	$24,176.00	$21,197.00	114.05%	$22,493.00	$23,025.00	97.69%	$26,234.00	$25,329.00	103.57%	$24,803.00	$23,374.00	106.11%	$20,278.00	$25,394.00	79.85%
	EDUCATION	$23,553.00	$23,018.00	102.32%	$27,746.00	$23,771.00	116.72%	$25,953.00	$24,064.00	107.85%	$25,332.00	$22,352.00	113.33%	$26,422.00	$21,495.00	122.92%
GERMANY	CONSUMER	$24,294.00	$23,905.00	101.63%	$21,217.00	$20,282.00	104.61%	$24,608.00	$22,740.00	108.21%	$28,690.00	$26,435.00	108.53%	$25,037.00	$23,081.00	108.47%
	EDUCATION	$21,840.00	$19,891.00	109.80%	$25,888.00	$23,787.00	108.83%	$23,894.00	$21,899.00	109.11%	$26,370.00	$23,082.00	114.24%	$24,160.00	$26,452.00	91.34%

Output 17. Using Multiple Conditions to Perform Trafficlighting

The CHAIR cells for CANADA were not given a different background color because the Pct of Predict is under 100%. However, the other cells do have the appropriate background colors. In addition, the Pct of Predict is given a green foreground color when the Pct of Predict is over 100% for the other data cells.

The heart of the COMPUTE block, where the multiple conditions are tested, is in this IF statement. The absolute columns _c5_, _c14_, and _c17_ are given a foreground of green when their Pct of Predict is greater than 100% (or 1.0). However, that code is simpler and is not shown here. We want to focus only on highlighting based on a complex condition, such as the ELSE condition below for CANADA.

```
if altcountry in ('GERMANY', 'U.S.A.') then do;
    if _c8_ gt 1.0 then
              call define('_c8_','style',
              'style={foreground=green background=white fontweight=bold}');
    else call define('_c8_','style',
              'style={foreground=black background=white fontweight=medium}');
    if _c11_ gt 1.0 then call define('_c11_','style',
              'style={foreground=green background=white fontweight=bold}');
    else call define('_c11_','style',
              'style={foreground=black background=white fontweight=medium}');
```

```
end;
else if altcountry = 'CANADA' and (_c8_ gt 1.0 or _c11_ gt 1.0) then do;
** only want to highlight CHAIR and DESK for Canada when PctCalc gt 100%;
         svar8 = 'style={background=' || put(_c8_,ovrpct.)||'}';
         svar11 = 'style={background=' || put(_c11_,ovrpct.)||'}';
         call define('_c6_','style',svar8);
         call define('_c7_','style',svar8);
         call define('_c8_','style',svar8);
         call define('_c9_','style',svar11);
         call define('_c10_','style',svar11);
         call define('_c11_','style',svar11);
 end;
```

Refer to the program in the ZIP file for the entire code. Note how the code for trafficlighting is applied differently for GERMANY and USA. versus how the comparison for CANADA and a test for _c8_ and _c11_ are used in the ELSE condition. This ensures that only the columns for CHAIR and DESK will get the desired trafficlighting

CONCLUSION

The possibilities with PROC REPORT are almost endless. What initially seems like report hurdles ends up being unique ways to use PROC REPORT to create unique reports. Whether you need to “transpose” without using PROC TRANSPOSE or use character variables with an ACROSS item, or perform difficult trafficlighting, or use CALL DEFINE, the bottom line is that using ACROSS with PROC REPORT is easier than you think and well worth the investment of time to learn how to sail over the ACROSS hurdle.

REFERENCES

Booth, Allison McMahill. 2010. "Evolve from a Carpenter’s Apprentice to a Master Woodworker: Creating a Plan for Your Reports and Avoiding Common Pitfalls in REPORT Procedure Coding." *Proceedings of the SAS Global Forum 2010 Conference.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings10/133-2010.pdf.

Booth, Allison McMahill. 2011. "Beyond the Basics: Advanced REPORT Procedure Tips and Tricks Updated for SAS® 9.2." *Proceedings of the SAS Global Forum 2011 Conference.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings11/246-2011.pdf.

Booth, Allison McMahill. 2012. "PROC REPORT Unwrapped: Exploring the Secrets behind One of the Most Popular Procedures in Base SAS® Software." *Proceedings of the Pharmaceutical Industry 2012 SAS Users Group Conference*. Cary, NC: SAS Institute Inc. Available at http://www.pharmasug.org/proceedings/2012/TF/PharmaSUG-2012-TF20-SAS.pdf.

McMahill, Allison. 2007. "Advanced PROC REPORT Tips and Tricks." *Proceedings of the SAS Global Forum 2007 Conference.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/rnd/papers/sgf07/sgf2007-report.pdf.

SAS Institute Inc. 2008. "The REPORT Procedure: Getting Started with the Basics." Available at http://support.sas.com/resources/papers/ProcReportBasics.pdf.

SAS Institute Inc. 2008. "Using Style Elements in the REPORT and TABULATE Procedures." Available at http://support.sas.com/resources/papers/stylesinprocs.pdf.

Zender, Cynthia L. 2008. "Creating Complex Reports." *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. Available at http://www2.sas.com/proceedings/forum2008/173-2008.pdf.

Zender, Cynthia L., and Allison M. Booth. 2013. "Turn Your Plain Report into a Painted Report Using ODS Styles." *Proceedings of the SAS Global Forum 2013 Conference.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings13/366-2013.pdf.

ACKNOWLEDGMENTS

The author would like to thank Michele Ensor, Linda Jolley, and Allison Booth, who generously read the paper for content and made suggestions for improving the paper. In addition, thanks are due to Caroline Brickley, who edited the paper and made it more readable. Finally, many thanks are due to Tim Hunter, for help with PROC REPORT and the PROC REPORT team in Tech Support for answering endless questions about why things work the way they work with PROC REPORT. All song snippets in the section headers are from song lyrics about helping, jumping, and flying that became the “soundtrack” for this paper. (And, yes, my middle name is Louise.)

CONTACT INFORMATION

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

Cynthia Zender
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Cynthia.Zender@sas.com
http://www.sas.com