

Paper 202-27

Report Macro - A Tool to Generate Flexible Summary Reports

Diane E. Foose, GlaxoSmithKline, Collegeville, Pa
copyright © GSK, 2002

Abstract

When creating output, you can not always anticipate the number of columns needed in your summary tabulation or that each page of the report may vary on the number of columns used. This may cause you to program many variations of the same style of report for each slightly different output.

This paper will present the concept of using a SAS® macro as an easy-to-use solution to this problem. Considerations for how to use this strategy for reporting your data will be discussed. An implementation of this concept, with examples, will also be presented.

Introduction

Many requests for summarized output may come your way, but the majority of the output is basically just a table with cells filled in. Your output may be divided by age group, gender, geographic location, treatment, etc. It usually has a count for each cell with possibly an associated percentage. Other than that they are all similar. If you can create one record for each cell of your table, then you can use this macro to produce your summary.

Design Specifications

The report macro was designed initially to create a tabulation for laboratory data that varied in the number of columns of output per lab test. Each page of the final report summarized one laboratory test. Proc Report handles this very nicely with the *across* function. All that is needed is one record of output for each cell in the final output and Proc Report will determine how many columns are needed for each page. After seeing this work, I thought it might also work for other reports I needed to produce. Instead of creating a Proc Report step for each table, I decided to make one macro that handles all types of data and will output it correctly.

In order for this macro to work, you need to pass in the information on the page, column and row variables. The macro assumes that each record

contains the necessary variables to hold the page, column and row information and the actual cell values. The actual counting is accomplished in a separate program and then the report macro is called and the variable names are passed to it to create the output.

Options for macro

The options for the output are as follows:

- 1-4 Paging Variables
- 1-4 Row Variables
- 1-2 Column Variables
- 1-2 Cell Variables

For each of these variables you can specify the width, format, label, and an associated order variable for the column and row variables. At minimum, the variables you need for this macro are the row, column and cell variables along with the data set. If you use the defaults given and only have one column variable (*col_code*), a column format (*colfmt.*) and one cell variable (*value*), then all you need to pass to this macro are the row variable, row variable count and the data set being used.

I found the number of variables to be enough to cover all types of tables that I needed to produce. This macro could be expanded to add even more columns, if necessary.

Examples

The next three samples will demonstrate various uses of this macro. The input reflects one record per cell and the output reflects the use of various options within the macro.

In the below sample input and output #1, there are no paging variables. The "Characteristic" category (*param*) and the groupings within the category (*class*) are the row variables. The "Regimen" variable (*col_code*) is the column variable. The combined count and percent values (*value*) is the cell value.

SAMPLE INPUT #1

CLASSORD	CLASS	PARORD	PARAM	COL_CODE	VALUE
< 60 KG	< 60 KG	05	WEIGHT CATEGORY	1	177 (15.7%)
< 60 KG	< 60 KG	05	WEIGHT CATEGORY	2	394 (15.7%)
< 65	< 65	03	AGE CATEGORY	1	853 (57.8%)
< 65	< 65	03	AGE CATEGORY	2	1809 (59.1%)
CAUCASIAN	CAUCASIAN	02	RACE	1	1107 (96.1%)
CAUCASIAN	CAUCASIAN	02	RACE	2	2427 (96.7%)
MALE	MALE	01	SEX	1	802 (71.0%)
MALE	MALE	01	SEX	2	1742 (69.4%)
01	N	04	AGE (YRS)	1	1129
01	N	04	AGE (YRS)	2	2511
01	N	06	WEIGHT (KG)	1	1120
01	N	06	WEIGHT (KG)	2	2511
02	MEAN	04	AGE (YRS)	1	59.5
02	MEAN	04	AGE (YRS)	2	59.5A
02	MEAN	06	WEIGHT (KG)	1	74.7
02	MEAN	06	WEIGHT (KG)	2	75.1
03	STD	04	AGE (YRS)	1	13.2
03	STD	04	AGE (YRS)	2	13.8
03	STD	06	WEIGHT (KG)	1	15.5
03	STD	06	WEIGHT (KG)	2	15.8
04	MEDIAN	04	AGE (YRS)	1	62.0
04	MEDIAN	04	AGE (YRS)	2	62.0
04	MEDIAN	06	WEIGHT (KG)	1	74.0
04	MEDIAN	06	WEIGHT (KG)	2	74.0
05	MIN	04	AGE (YRS)	1	10
05	MIN	04	AGE (YRS)	2	17
05	MIN	06	WEIGHT (KG)	1	34
05	MIN	06	WEIGHT (KG)	2	30
06	MAX	04	AGE (YRS)	1	94
06	MAX	04	AGE (YRS)	2	92
06	MAX	06	WEIGHT (KG)	1	140
06	MAX	06	WEIGHT (KG)	2	196
60-80 KG	60-80 KG	05	WEIGHT CATEGORY	1	595 (52.7%)
60-80 KG	60-80 KG	05	WEIGHT CATEGORY	2	1280 (51.0%)
65 - 74	65 - 74	03	AGE CATEGORY	1	376 (32.3%)
65 - 74	65 - 74	03	AGE CATEGORY	2	817 (32.5%)

Sample Input #2

TRX	TEST_PRM	EO2_ORD	COL_CODE	TIME_PT	TIME_ORD	BASELINE	BASE_ORD	DENOM	VALUE
141	PLACERO FR (MSEC) /	3	4	TOTAL ON-THERAPY	1	NORMAL (<200 MSEC)	2	1007	2 (0.2%)
142	PLACERO FR (MSEC) /	3	3	TOTAL ON-THERAPY	1	NORMAL (<200 MSEC)	2	1007	60 (6.0%)
143	PLACERO FR (MSEC) /	3	2	TOTAL ON-THERAPY	1	NORMAL (<200 MSEC)	2	1007	945 (93.8%)
144	PLACERO FR (MSEC) /	3	4	TOTAL ON-THERAPY	1	HIGH (200-249 MSEC)	3	61	1 (1.6%)
145	PLACERO FR (MSEC) /	3	3	TOTAL ON-THERAPY	1	HIGH (200-249 MSEC)	3	61	13 (21.3%)
146	PLACERO FR (MSEC) /	3	2	TOTAL ON-THERAPY	1	HIGH (200-249 MSEC)	3	61	6 (85.7%)
147	PLACERO FR (MSEC) /	3	4	TOTAL ON-THERAPY	1	CONCERN (>=250 MSEC)	4	7	6 (85.7%)
148	PLACERO FR (MSEC) /	3	3	TOTAL ON-THERAPY	1	CONCERN (>=250 MSEC)	4	7	1 (14.3%)
149	PLACERO FR (MSEC) /	3	2	TOTAL ON-THERAPY	1	CONCERN (>=250 MSEC)	4	7	0
150	PLACERO FR (MSEC) /	3	4	TOTAL ON-THERAPY	1	MISSING	8	7	1 (14.3%)
151	PLACERO FR (MSEC) /	3	3	TOTAL ON-THERAPY	1	MISSING	8	7	1 (14.3%)
152	PLACERO FR (MSEC) /	3	2	TOTAL ON-THERAPY	1	MISSING	8	7	5 (71.4%)
153	PLACERO FR (MSEC) /	3	4	TOTAL ON-THERAPY	1	TOTAL	9	1082	10 (0.9%)
154	PLACERO FR (MSEC) /	3	3	TOTAL ON-THERAPY	1	TOTAL	9	1082	109 (10.1%)
155	PLACERO FR (MSEC) /	3	2	TOTAL ON-THERAPY	1	TOTAL	9	1082	963 (89.0%)
156	PLACERO FR (MSEC) /	3	4	ENDPOINT	2	NORMAL (<200 MSEC)	2	1015	1 (0.1%)
157	PLACERO FR (MSEC) /	3	3	ENDPOINT	2	NORMAL (<200 MSEC)	2	1015	30 (3.0%)
158	PLACERO FR (MSEC) /	3	2	ENDPOINT	2	NORMAL (<200 MSEC)	2	1015	994 (96.9%)
159	PLACERO FR (MSEC) /	3	4	ENDPOINT	2	HIGH (200-249 MSEC)	3	61	0
160	PLACERO FR (MSEC) /	3	3	ENDPOINT	2	HIGH (200-249 MSEC)	3	61	39 (63.9%)
161	PLACERO FR (MSEC) /	3	2	ENDPOINT	2	HIGH (200-249 MSEC)	3	61	12 (36.1%)
162	PLACERO FR (MSEC) /	3	4	ENDPOINT	2	CONCERN (>=250 MSEC)	4	7	5 (71.4%)
163	PLACERO FR (MSEC) /	3	3	ENDPOINT	2	CONCERN (>=250 MSEC)	4	7	2 (28.6%)

Sample Output #1

DEMOGRAPHIC CHARACTERISTICS
OF PATIENTS IN UNCONTROLLED STUDIES

Regimen A (N= 1043)		
CHARACTERISTIC		
SEX	MALE	815 (78.1%)
	FEMALE	228 (21.9%)
RACE	CAUCASIAN	1011 (96.9%)
	BLACK	12 (1.2%)
	OTHER	20 (1.9%)
AGE CATEGORY	< 65	503 (48.2%)
	65 - 74	444 (42.6%)
	>= 75	96 (9.2%)
AGE (YRS)	N	1043
	MEAN	63.6
	STD	8.9
	MEDIAN	65.0
	MIN	39
	MAX	84

Sample Output #2, Page 1

TRANSITIONS FROM BASELINE IN ECG PARAMETERS OBTAINED AT TROUGH
FOR PATIENTS IN CONTROLLED STUDIES

PARAMETER=PR (MSEC) / TIMEPOINT-TOTAL ON-THERAPY	BASELINE	NUMBER OF PATIENTS TESTED		DOUBLE-BLIND		
		NORMAL	HIGH	CONCERN		
REGIMEN A	NORMAL (<200 MSEC)	1007	945 (93.8%)	60 (6.0%)	2 (0.2%)	
	HIGH (200-249 MSEC)	61	13 (21.3%)	47 (77.0%)	1 (1.6%)	
	CONCERN (>=250 MSEC)	7	0	1 (14.3%)	6 (85.7%)	
	MISSING	7	5 (71.4%)	1 (14.3%)	1 (14.3%)	
	TOTAL	1082	963 (89.0%)	109 (10.1%)	10 (0.9%)	
REGIMEN B	NORMAL (<200 MSEC)	2149	2007 (93.4%)	139 (6.5%)	3 (0.1%)	
	HIGH (200-249 MSEC)	148	37 (25.0%)	99 (66.9%)	12 (8.1%)	
	CONCERN (>=250 MSEC)	15	1 (6.7%)	5 (33.3%)	9 (60.0%)	
	MISSING	28	21 (75.0%)	5 (17.9%)	2 (7.1%)	
	TOTAL	2340	2066 (88.3%)	248 (10.6%)	26 (1.1%)	

The following is the macro call used to create output #1:

```
%report(data = p_dem,
colbyn = 1,
colby1 = col_code,
rowbyn = 2,
rowby1 = param,
rowby1o = parord,
rowby1l = CHARACTERISTIC,
rowby2 = class,
rowby2o = classord);
```

Sample Output #2, Page 2

TRANSITIONS FROM BASELINE IN ECG PARAMETERS OBTAINED AT TROUGH
FOR PATIENTS IN CONTROLLED STUDIES

PARAMETER=ATRIAL RATE (BPM) / TIMEPOINT-TOTAL ON-THERAPY	BASELINE	NUMBER OF PATIENTS TESTED		DOUBLE-BLIND				
		LOW CONCERN	LOW	NORMAL	HIGH	HIGH CONCERN		
REGIMEN A	LOW CONCERN (<50 BPM)	16	9 (56.3%)	4 (25.0%)	3 (18.8%)	0	0	
	LOW (50-59 BPM)	107	14 (13.1%)	71 (66.4%)	20 (18.7%)	4 (3.7%)	1 (0.9%)	
	NORMAL (60-99 BPM)	921	6 (0.7%)	154 (16.7%)	697 (75.7%)	61 (6.6%)	7 (0.8%)	
	HIGH (100-120 BPM)	39	0	0	13 (33.3%)	23 (59.0%)	3 (7.7%)	
	HIGH CONCERN (>120 BPM)	0	0	0	0	0	0	
	MISSING	3	0	0	3 (100.0%)	0	0	
	TOTAL	1086	29 (2.7%)	229 (21.1%)	736 (67.8%)	88 (8.1%)	11 (1.0%)	
REGIMEN B	LOW CONCERN (<50 BPM)	37	24 (64.9%)	11 (29.7%)	2 (5.4%)	1 (2.7%)	0	
	LOW (50-59 BPM)	289	41 (14.2%)	177 (61.2%)	70 (24.2%)	5 (1.7%)	1 (0.3%)	
	NORMAL (60-99 BPM)	1957	9 (0.5%)	280 (14.3%)	1515 (77.4%)	153 (7.8%)	7 (0.4%)	
	HIGH (100-120 BPM)	57	0	0	23 (40.4%)	32 (56.1%)	2 (3.5%)	
	HIGH CONCERN (>120 BPM)	4	0	1 (25.0%)	1 (25.0%)	0	2 (50.0%)	
	MISSING	21	2 (9.5%)	8 (38.1%)	10 (47.6%)	1 (4.8%)	0	
	TOTAL	2365	76 (3.2%)	477 (20.2%)	1621 (68.5%)	192 (8.1%)	12 (0.5%)	

In the above output #2 samples, "Parameter" (*test_prm*) and "Timepoint" (*time_pt*) are the paging variables. "Regimen" (*trx*), "Baseline" (*baseline*) and "Number of Patients Tested" (*denom*) are the row variables. The "double-blind" variable (*col_code*) is the column variable. The count and percent values are combined into one variable (*value*) and is the cell variable. The first sample output page has three distinct values for the column variable -- Normal, High and Concern. The second sample output page has five distinct values for the column variable - Low

Concern, Low, Normal, High and High Concern. For each combination of the paging variables Proc Report will determine how many columns are needed and display only these columns.

Below is the macro call used to create output #2:

```
%report(data=p_trans,
        pagebyn = 2,
        pageby1= test_prm,
        pageby2 = time_pt,
        pageby1f = $testprm.,
        pageby2f = $timept.,
        rowbyn = 3,
        rowby1 = trx,
        rowby1w = 10,
        rowby2 = baseline,
        rowby2l = BASELINE,
        rowby2o = base_ord,
        rowby2w = 29,
        rowby3 = denom,
        rowby3l = NUMBER
        OF?PATIENTS?TESTED,
        rowby3w = 9,
        rowbyn = 3,
        colbyn = 1,
        colby1 = col_code,
        colby1f = colfmt.,
        colfmt = $flagfmt.,
        splitch = ?);
```

variables.

The following is the macro call used to create output #3:

```
%report(data=end,
        valuen=2,
        value1=endcnt,
        value1l=N,
        value2=endpct,
        value2l=%,
        colbyn=1,
        colby1=term_c,
        colby1f=termfmt.,
        colby1w=20,
        colby1l=%str(--Term Reason--),
        rowbyn=1,
        rowby1=relday,
        rowby1l=Time Interval (Days),
        rowby1w=15,
        splitch=_);
```

Conclusion

By using this adaptable macro, you can display any type of data with any combination of page, row, column and cell variables. It will also provide a consistent 'look' for all of your output. Hopefully, this can make for easier programming and flexible reports.

Acknowledgements

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.

Contact Information

Diane E. Foose
 GlaxoSmithKline
 1250 S. Collegeville Road, OK1105
 Collegeville, Pa 19426-0989
 (610)917-7168 (W)
 e-mail: diane.foose-1@gsk.com



Sample Input #3			
TERM_C	RELDAY	ENDCNT	ENDPCT
1	Total Patients	1	100
1	> 14 Days	1	100
7	Total Patients	4	100
7	6 - 14 Days	3	75
7	> 14 Days	1	25
99	Total Patients	5	100
99	6 - 14 Days	3	60
99	> 14 Days	2	40

Sample Output #3						
Cumulative Percentage Of Patient Withdrawals From Study Overall Over Time						
Time Interval (Days)	----- Term Reason -----					
	AE		Other		Total	
	N	%	N	%	N	%
Total Patients	1	100	4	100	5	100
6 - 14 Days	0	0	3	75	3	60
> 14 Days	1	100	1	25	2	40

In the above sample output #3, there are no paging variables. "Time Interval" (*relday*) is the row variable and the "Term Reason" (*term_c*) is the column variable. 'N' (*endcnt*) and '%' (*endpct*) are the cell

THE REPORT MACRO

```

/*****
/* Program:      report.sas
/* Author:      Diane Foose
/* Date:       29-Jan-2001
/*
/* Abstract:    To generate summary reports with either categorical or
/*              continuous variables
/*
/* Invocation:  %report(data=);
/*              ex. %report(data=p_dem p_pulm,rowbyn=1,rowby1=grp);
/*
/* Parameters:  The following parameters are a part of this macro, the ones
/*              with an asterisk (*) are mandatory. The rest are optional
/*              or the defaults can be used.
/*
/*      *data      = datasets needed to produce all data for output
/*                  (more than one dataset may be entered, but they
/*                  must all have the same variable names)
/*
/*      pagebyn    = Number of pageby variables
/*      pageby1    = Name of first Pageby variable
/*      pageby1f   = Format variable for first Pageby variable
/*      pageby2    = Name of second Pageby variable
/*      pageby2f   = Format variable for second Pageby variable
/*      pageby3    = Name of third Pageby variable
/*      pageby3f   = Format variable for third Pageby variable
/*      pageby4    = Name of fourth Pageby variable
/*      pageby4f   = Format variable for fourth Pageby variable
/*      valuen     = Number of cell value variables
/*      value1     = Name of first cell variable
/*                  (default=value)
/*      value2     = Name of second cell variable
/*      value1l    = Label for first cell value (optional)
/*      value2l    = Label for second cell value (optional)
/*      colbyn     = Number of column variables (default=1)
/*      colby1     = first column code variable name (default=col_code)
/*                  (used as Across variable, data should be formatted)
/*      colby1w    = Column code width for printing (default=20)
/*      colby1l    = Column spanning label (if desired)
/*      colby2     = Second column code variable name
/*      colby2w    = Second Column code width for printing
/*      colby2l    = Column spanning label (if desired)
/*      *rowbyn    = Number of rowby variables
/*      *rowby1    = Name of first rowby variable
/*      rowby2     = Name of second rowby variable
/*      rowby3     = Name of third rowby variable
/*      rowby4     = Name of fourth rowby variable
/*      rowby1l    = Label for first rowby variable
/*      rowby2l    = Label for second rowby variable
/*      rowby3l    = Label for third rowby variable
/*      rowby4l    = Label for fourth rowby variable
/*      rowby1o    = Ordering variable for first rowby variable
/*      rowby2o    = Ordering variable for second rowby variable
/*      rowby3o    = Ordering variable for third rowby variable
/*      rowby4o    = Ordering variable for fourth rowby variable
/*      rowby1f    = Format variable for first rowby variable
/*      rowby2f    = Format variable for second rowby variable

```

```

/*          rowby3f  = Format variable for third rowby variable          */
/*          rowby4f  = Format variable for fourth rowby variable         */
/*          rowby1w  = variable width for first rowby variable          */
/*          rowby2w  = variable width for second rowby variable         */
/*          rowby3w  = variable width for third rowby variable          */
/*          rowby4w  = variable width for fourth rowby variable         */
/*          break    = name of break variable if other than rowby1o is  */
/*                   desired                                           */
/*          spltch   = Split character for proc report output           */
/*                                                                */
/* Formats:      None                                                  */
/*                                                                */
/* Input:        data is expected to be in the following layout:      */
/*               prot pid &rowby &rowbyo &pageby &value &colby       */
/*               (example: prot pid class classord param paramord value colcode */
/*               - each row is one cell to be printed)                 */
/*               where you can have :                                  */
/*               1 to 2 column variables                             */
/*               1 to 2 cell variables                               */
/*               1 to 4 rowby variables (and row order variables)     */
/*               1 to 4 pageby variables                             */
/*                                                                */
/* Output:       *.lst                                                  */
/*****

%macro report(data=,          valuen=1, value1=value, value2=, value1l=, value2l=,
                colbyn=1, colby1=col_code, colby1f=colfmt., colby1w=20, colby1l=,
                colby2=, colby2f=, colby2w=, colby2l=,
                rowbyn=, rowby1=, rowby2=, rowby3=, rowby4=,
                rowby1l=, rowby2l=, rowby3l=, rowby4l=,
                rowby1w=, rowby2w=, rowby3w=, rowby4w=,
                rowby1o=, rowby2o=, rowby3o=, rowby4o=,
                rowby1f=, rowby2f=, rowby3f=, rowby4f=,
                pagebyn=0, pageby1=, pageby2=, pageby3=, pageby4=,
                pageby1f=, pageby2f=, pageby3f=, pageby4f=,
                break=, spltch=_);

*****
*   combine all datasets to be analyzed                               *;
*   Add numeric dummy variable to make across analysis work in proc report *;
*****
data final;
    set &data;
    dummy = 1;
run;

proc sort data=final;
    by %do a = 1 %to &pagebyn; &&pageby&a %end;
        %do b = 1 %to &colbyn; &&colby&b %end;
    ;

*****
*** output final report                                             *;
*****
option nobyline;

proc report data=final headline headskip nowindows split="&spltch";

```

```

%if &pagebyn > 0 %then %do;
  by %do a = 1 %to &pagebyn; &&pageby&a %end; ;
  %do b = 1 %to &pagebyn;
    %if "&&pageby&b.f" ^= "" %then format &&pageby&b &&pageby&b.f;;
  %end;
%end;
column %do r = 1 %to &rowbyn; &&rowby&r.o &&rowby&r %end;
  %if &colbyn = 1 %then &colby1;
  %else (&colby1,&colby2);
  ,%if &valuen = 1 %then &value1;
  %else (&value1 &value2);
  dummy;
%do k = 1 %to &rowbyn;
  %if "&&rowby&k.o" ^= "" %then          /* if rowby order variable not missing */
  define &&rowby&k.o / group noprint;;
  define &&rowby&k / group flow "&&rowby&k.l"
    %if "&&rowby&k.w" ^= "" %then width=&&rowby&k.w;
    %if "&&rowby&k.f" ^= "" %then format=&&rowby&k.f;;
%end;
%do d = 1 %to &colbyn;
  define &&colby&d / across order=internal
    %if "&&colby&d.w" ^= "" %then width=&&colby&d.w
    %if "&&colby&d.f" ^= "" %then format=&&colby&d.f;
    %if "&&colby&d.l" ^= "" %then "-&&colby&d.l-";
    %else ' ';;
%end;
%do v = 1 %to &valuen;
  define &&value&v / display "&&value&v.l";
%end;
  define dummy / analysis '' noprint;

  break after %if &break ^= %then &break; %else %if "&rowby1o" ^= "" %then &rowby1o;
%else &rowby1; / skip;
  %if &pagebyn > 0 %then %do;
    title10
    %do p = 1 %to &pagebyn;
      #byvar&p= #byval&p
    %end;
  ;
%end;
run;

%mend report;

```