Paper PO15

Program Templates for Report Generation

Roger DeAngelis , AMGEN, Thousand Oaks, CA

ABSTRACT

A Program Template is a minimum sequence of SAS procedures, datasteps, and user formats, which can be easily modified to produce a wide range of reports. A Program Template with four characteristic properties is presented. First, the rows of the report have major and minor groupings; for example, “Gender” could be a major grouping, while “Male” and “Female” would be minor groupings. Second, the columns represent arbitrary subsets of the observations; in a clinical environment, “Placebo” and “Treatment” are typical columns. Third, SAS title and footnote statements are reserved for Microsoft Word headings and footers, while Report titles and footnotes are part of the report body. Finally, two sets of page numbers are supported: one set for Microsoft Word footer space and another for the body of the report. Well designed Program Templates give the programmer much more control over the layout of reports

INTRODUCTION

A program template consisting of six procedures is given. This paper focuses is on data structure, data transformation and formatting(proc format) rather than complex reporting options.

BRIEF DESCRIPTION OF INPUTS, PROCESS AND OUTPUTS(IPO) FOR REPORTING

Methods presented here apply to a wide range of industries, such as marketing, banking and pharmaceuticals, however for clarity this paper will focus on pharmaceutical reporting.

INPUTS

The input variables are classified into three logical groups. First, identification variables provide a unique identifier for each observation, patient number(Pat) and treatment number(Trt) are identification variables. Treatment has three levels, Placebo(1), Zyplex(2) and Total(3). The Total group is the result of replicating all data and assigning 3 to the treatment number. Second, nominal(discrete) variables classify subjects into a limited number of groups. Sex, ethnicity, country and state are examples of nominal variables. Finally, continuous ordinal variables provide input for ‘proc univariate’. Examples of ordinal variables are height, weight and white blood cell.

DATA CONSIDERATIONS

Numeric variables are used for classification(Sex) and factual data(Weight). Numeric classification variables, such as 0/1, are more flexible then character strings such as 'Male' and 'Female'. These 0/1 variables provide order and can be used directly in Boolean algebra. For instance diabetic males can be selected using the clause 'where Sex*Dia or Where Sex*Dia=1'. If Females are assigned 1 then the sum of Sex will be the number of females and the mean of sex will give the fraction of patients that are female. Always take care when assigning numeric levels to classification variables, if you want female statistics to appear after males use 1 for females and 0 for males. Do not attach user formats to input SAS table, use simple num and $n default formats Do not use multiple missing value levels, use '.' for all missing values. User formats provide descriptions for the classification data.

The raw input data, table Fab_DemRaw.sas7bdat, has the following structure.(The data has been split into two sections for readability, the ordinal and nominal data are on same line.)

```
          Ordinal Variables
          ================================================
 Pat Trt Wgt Hgt Bpd Bps Qol Wbc Rbc Age Bun Cre Ast Alt Glu

   1  1  314 516 480 986 915 895 397 385 628 698 434 776 868
   2  1   69 546  31 125 733 392 687  11 652 438 362 726 184
   3  1  852 512 297  93 447 338 897 972 286 348 518 158 379
   4  1  814 367 717 978 644 653 517 959 543 747 723 977 388
   5  1  315 731 799 955 286 518 519 804 913 979 363 612 520
   6  1  606 464  45 365  16 417 971 368  73 343 480 176 775
   7  1  716 538 593 731 848 981 264 808   8 894  55 528 847
   8  1  906 521 936 452 528 216 275 314 152 626 934 197  72


     Nominal Variables
```

Pat	Trt	Hiv	Sex	Usa	Itt	Enr	Pox	Ane	Smo	Dia	Inc	Occ	Edu	Bla	Mar	Kid	Flu
1	1	0	1	0	1	0	1	1	1	0	LWR	MAN	PHD	1	1	0	0
2	1	0	0	0	1	0	0	0	1	0	UPR	TEC	MAS	0	0	0	0
3	1	1	0	0	0	0	0	1	0	1	UPR	TEC	MAS	0	0	0	0
4	1	1	1	1	0	1	0	0	1	1	LWR	MAN	PHD	1	0	1	1
5	1	0	1	0	0	0	1	1	0	1	MED	LBR	BAC	1	0	1	1
6	1	1	1	1	0	1	0	1	1	1	UPR	TEC	MAS	0	1	0	1
7	1	0	0	0	0	1	1	0	0	1	LWR	MAN	PHD	0	1	0	1
8	1	0	1	0	1	1	0	0	0	0	MED	LBR	BAC	1	0	1	1

The contents of Fab_DemRaw are given below. This output is produced by macro Utl_ConMaxMin.sas. Note that all variables are numeric.

PROCESS

The following lines of SAS code are the core code for this paper. The code can be broken down into five procedures. First, the nominal(categorical) data is transposed. Second, univariate statistics are computed for all ordinal(numeric) data. Third, the ordinal univariate statistics are transposed. Fourth, SQL computes the counts and percents for the nominal data and combines these results with the univariate results. Fifth, the long and skinny output of Proc Sql is transposed with treatments as columns. Finally, proc report creates the clinical report.

A more detailed explanation follows. The first transpose operates on nominal data to create a normalized long and skinny data structure with variable names as data. The primary key is Pat(Patient), Trt(Treatment) and Variable name(ie Smo is Smoker 0/1). The final column contains the data associated with variable name. This data structure is input to the SQL processor. SQL can easily provide N(%) statistics using this data structure. Next the means procedure operates on ordinal data to create a very wide data structure. All possible univariate statistics are computed for each ordinal vaiable, in this case 2 + 28*13 or 366 columns are created. Since all variable names are three characters the columns will have names like Age_Mean, Age_Max or Age_Min. Finally we transpose the means output into a very long and skinny data structure which is identical to the output of first transpose. Next, SQL computes the N(%) statistics, column headings, union of N(%) statistics and univariate statistics, compound statistics like 'Min,Max' and applies the formats. The last transpose creates the treatment columns. Finally Proc Report is used like a 'Proc Print' to display the data. Details describing each line of code are given in the appendix.

LINES OF SAS CODE

The core code consists of a Proc Transpose, Proc Means, Proc Transpose, Proc Sql, Proc transpose and a Proc Report. Variable names, data classification variables and formats have been designed to minimize the code needed for the report, complete example is given in the appendix.

```
Proc Transpose Data=DemRaw Out=DemXpoNom ;     /* nominal data long and skinny */

Proc Means Data=DemRaw                          /* means ordinal data */

Proc Transpose Data=DemUnv Out=DemUnvLng;       /* univariate stats long and skinny*/

Proc Sql create Table DemPre .. From DemUnvLng  /* prep for a simple proc report */

Proc Transpose Data=DemPre Out=DemPreXpo;       /* short and fat treatments across */

Proc Report ...                                 /* dumb proc report */
```

OUTPUTS

Defining the table shell, defining the layout of each page, calculating the two sets of page numbers and defining column headings with the (N=XX) term are discussed below.

OUTPUT REPORT TABLE SHELL

The Output report has the generic form :

```
                  Grp-1    Grp-2    Total
                 (N=XX)    (N=XX)   (N=XX)
  Major Group
   Minor Group
```

A specific example of the output is:

```
Ajax Pharmaceuticals                          Protocol FAB
1750 Westwind Dr
Dallas TX 87654


    Table 1.1.1 Demographic Characteristics

                    Placebo     Zyplex      Total
                    (N=100)     (N=100)     (N=200)
      Sex
        Male         25(25%)    25(25%)     25(25%)
        Female       75(75%)    75(75%)     75(75%)

      Age(Yrs)[1]
        N            100        100         200
        Mean(Sd)     56(10)     58(11)      57(11)
        Median       52         56          55
        Min, Max     18,76      20,87       18,87


      White Blood Cell Count[2]
        N            100        100         200
        Mean(Sd)     56(10)     58(11)      57(11)
        Median       52         56          55
        Min, Max     18,76      20,87       18,87

    [1] Age at First Dose Date
    [2] Baseline is the average of all screening visits

    PGM: C:\Fab\Fab_Dem.sas
    OUT: C:\Fab\Fab_Dem.rtf              Page 1 of 17

PharmaSUG 2005
Phoenix AZ                                    Page 22 of 76
```

COLUMN HEADINGS

The column headings:

```
    Placebo     ZYPLEX      Total
    (N=100)     (N=100)     (N=100)
```

are computed using the following SQL clause.

```
  Select Resolve(
                '%Let _'!!Put(Trt,1.)!!
                 '=%Sysfunc(Putc(_'!!
                 Put(Trt,1.)!!',$Num2Trt.))#(N='!!
                 Put(Count(Trt),4.)!!');') as Dum
                )
  from DemRaw Group by Trt;
```

The mapping of treatment number to treatment description is given by:

```
  value $Num2Trt
     '_1'='Placebo'
     '_2'='Zyplex'
     '_3'='Total'
```

The resolve function above is executed three times, the arguments are:

```
%Let _1=Placebo#(N=100);
%Let _2=Xyplex#(N=100);
%Let _3=Total#(N=200);
```

The '#' sign is the 'proc report' split character.

These macro variables above are used for the column descriptions in 'Proc Report'. In this case column _1 will have the heading

```
Placebo
(N=100)
```

The macro variable “&_1” provides the heading for treatment column ‘_1’. In the RTF specification '\R\b \fs24 &_1' defines a bold 12pt bold font for the text 'Placebo#(N=100)'.

```
Define _1     / Display   "\R\b \fs24 &_1"  style(column)={just=c};
Define _2     / Display   "\R\b \fs24 &_2"  style(column)={just=c};
Define _3     / Display   "\R\b \fs24 &_3"  style(column)={just=c};
```

PAGE LAYOUT MAJOR ROWS

Formats provide a flexible technique for page layout. In clinical reporting the statistician provides a table shell that defines the contents of each page. Suppose page one is:

```
Table 1.1.1 Demographic Characteristics

                Placebo    Zyplex     Total
                (N=100)    (N=100)    (N=100)
  Gender
    Male         XX(XX%)   XX(XX%)    XX(XX%)
    Female       XX(XX%)   XX(XX%)    XX(XX%)

  Age(Yrs)¹
    Mean(Sd)     XX(XX)    XX(XX)     XX(XX)
    Median       XX        XX         XX
    Min, Max     XX,XX     XX,XX      XX,XX

  White Blood Cell Count²

    Mean(Sd)     XX(XX)    XX(XX)     XX(XX)
    Median       XX        XX         XX
    Min, Max     XX,XX     XX,XX      XX,XX

¹ Age at First Dose Date
² Baseline is the average of all screening visits

PGM: C:\Fab\Fab_Dem.sas
OUT: C:\Fab\Fab_Dem.rtf              Page 1 of 17
```

The major headings for the first page are Sex, Age and WBC in that order. The format Mjr2Des provides the order and description for the major rows. The first two digits of the description are the page number, while the next two provide the order within the page. The last two digits are reserved for the minor row order.

```
value $Mjr2Des  /* Major heading to page, row and description */

'Sex' = '010100 Gender'
'Age' = '010200 Age(Yrs)¹'
'Wbc' = '010300 White Blood Cell Count'  /*  3rd section on page 1 */

 .....

'Alt' = '120100 Alt Lab Value'           /* major headings page 12 */
'Glu' = '120200 Glucose'
```

PAGE LAYOUT MINOR ROWS (DISCRETE NOMINAL VARIABLES)

The order and description of minor nominal variables(i.e. Sex) is given by the variables value and user format, respectively. In this case 'Male' statistics would appear before 'Female'. In Version 8 format names like 'Sex2Mnr' (maps 0/1 to Male/Female) require three character variable names(I.e. Sex). In version 9 format names may be up to 31 characters, thus variable name Sex could be up to 28 characters. For readability three character variable names

are used in this paper.

```
Value  Sex2Mnr
         0='Male'
         1='Female';
```

CONTINUOUS ORDINAL VARIABLES

The order, presence and description of minor continuous variable statistics are given by the first two digits of the description, an 'X' for presence and the remaining text, respectively. In our case N, Mean(SD), Median, Minimum and Maximum would be selected in that order. The page and line number for major heading 'Weight' and minor heading 'N' is defined as the sum of the major ordering number(i.e. 100800) and the minor ordering number for 'N' (01) or 100801. Likewise the page and line number for the major heading 'Weight' and minor heading 'Mean(SD)' are given by 100800 plus 03 or 100803. Obviously, it is very easy to change the content of any page.

```
  value $Mnr2Des

/* Put an X in position 6 to include this statistic */

'N'             =  '01 X N'
'NMiss'         =  '02   Number Missing'
'Mean'          =  '03 X Mean(SD)'
'LCLM'          =  '04   Lower and Upper 95% CI on Mean'
'UCLM'          =  '05   UCLM'
'Median'        =  '06 X Median'
'StdDev'        =  '07   Standard Deviation'
'Var'           =  '08   Var'
'Min'           =  '09 X Minimum,Maximum'
'Max'           =  '10   Max'
'Range'         =  '11   Range'
'Q1'            =  '12   Q1'
'Q3'            =  '13   Q3'
'Q1Q3'          =  '14   Q1,Q3'
'QRange'        =  '15   Quartile Range'
'StdErr'        =  '16   Standard Error of Mean'
'Skew'          =  '17   Skew'
'CV'            =  '18   Coef of Variation 100*StdDev/Mean'
'Sum'           =  '19   Sum'
'CSS'           =  '20   CSS'
'USS'           =  '21   USS'
'P1'            =  '22   1st, 99th Percentile'
'P5'            =  '23   5th, 95th Percentile'
'P10'           =  '24   10th, 90th Percentile'
'P90'           =  '25   P90'
'P95'           =  '26   P95'
'P99'           =  '27   P99'
'Probt'         =  '28   Probt'
't'             =  '29   t'
Other           =  '00    '
;
```

PAGE NUMBERING

MICROSOFT HEADER AND FOOTER SPACE PAGE NUMBERING

The RTF specification provides control words for page numbering, an example is given below. There is a bug in Microsoft Word, when you open a multi-page RTF file in Microsoft Word the first page will be 'Page 1 of 1'. However if you print the document or click on the header space the first page will display 'Page 1 of 13'.

```
title
 .j=l 'National Association of Free Thinkers'
 .j=r 'Page ^{PAGEOF}';
 .j=l "Las Vegas Nevada 2004^R/RTF'{\line}'";

footnote .j=l
"Journal of the Association of Free Thinkers Volume 11 November 2004^R/RTF'{\line}'"
"AMGEN 1 Biotech Court Thousand Oaks CA 91320";
```

PAGE NUMBERING IN THE BODY OF THE REPORT

The following code creates a right justified 'Page n of m' in the body of the report before the footnotes. This method also provides for an unlimited number of footnotes without the footnote gaps in many pharmaceutical reports.

There is a restriction in 'Proc Report', all line statements in a compute block must have the same justification. Since the 'Page n of m' must be right justified and footnotes left justified, two compute blocks and two page break variables are required.

In the first compute block below, a solid line is drawn from the left margin to right margin (^S={}\brdrt\brdrs) followed by a right justified 'Page n of m'.

The second compute block left justifies footnotes after the right justified 'Page n of m'. It may not be possible to write 'Page n of m' after the footnotes. The footnotes would have to be stored with the input data to get 'Page n of m' after footnotes. Actually, it is more flexible to store footnotes in the data.

CODE TO PRODUCE PAGE N OF M IN THE BODY OF THE REPORT

```
Break After Bdy / ;
Compute After Fot /style={just=r font_size=8pt};
  Hdr= compbl('Page '!!put( Fot, 2.)!!" of %Sysfunc(compress(&MaxPge.))");
  line  "^S={}\brdrt\brdrs";
  Line   Hdr $96.;
endcomp;
```

CODE FOR THE REST OF THE FOOTNOTES:

```
Break After Fot / suppress page;
Compute After _page_  / style={just=l font_size=8pt font_style=italic};
  line  "PGM: c:\Tut\&Pgm..sas";
  line  "OUT: &RtfOt1 &DatTym.";
endcomp;
```

HYPERLINK FOR SAMPLE DATA AND PROGRAM

http://homepage.mac.com/magdelina/.Public/utl.html

Table 1.1 Complex Demographic Table Produced by a User Modifiable SAS Template
Second Copy to Show Multiple Reports and Two sets of Page Numbers

Statistic	Valtrx (N= 161)	Placebo (N= 161)	Aspirin (N= 161)	Codeine (N= 161)	Vertex (N= 161)	All (N= 805)
Intend to Treat						
Could Not Treat	68(42.2%)	90(55.9%)	95(59.0%)	75(46.6%)	69(42.9%)	397(49.3%)
Treated	93(57.8%)	71(44.1%)	66(41.0%)	86(53.4%)	92(57.1%)	408(50.7%)
Enrollment						
Enrolled	87(54.0%)	76(47.2%)	92(57.1%)	74(46.0%)	86(53.4%)	415(51.6%)
Not Enrolled	74(46.0%)	85(52.8%)	69(42.9%)	87(54.0%)	75(46.6%)	390(48.4%)
Age						
N	161	161	161	161	161	805
Mean(SD)	(442.49,282.55)	(513.73,273.20)	(490.65,275.91)	(499.12,273.53)	(506.41,281.61)	(490.48,277.84)
Median	389.00	555.00	473.00	526.00	521.00	496.00
Minimum,Maximum	(2.00,982.00)	(13.00,980.00)	(5.00,987.00)	(1.00,996.00)	(24.00,999.00)	(1.00,999.00)

PGM: c:Pgm..sas
OUT: c:_DemRptNew.rtf 14JAN05:09:09:47

CONCLUSIONS

This paper describes a minimum sequence of SAS procedures(six procedures) yielding a very complex eleven page report. Instead of focusing on multiple data steps to transform arbitrary variable names and values, the focus was on integrating variable names, formats and variable types to obtain minimum SAS code. The templates, formats, dual page numbering techniques and dual headers and footers are applicable to general reporting.

The key points are:

1. Convert the input tables into very long and skinny tables as soon as possible. The first transpose on discrete variables yields:

Pat	Trt	_NAME_	_LABEL_	COL1
1	1	Inc	Income Level Lwr, Med , Upr	2
1	1	Edu	Bacalaurate Masters Phd	2
1	1	Pet	Type of Family Pet	2
1	1	Occ	Occupation Man, Tec, Lbr	2
1	1	Hiv	Aids 0-no 1-yes	1
1	1	Flu	Ever have the Flu	1
1	1	Sex	Gender Male Female	1

The second transpose on continuous variables yields:

Trt	_NAME_	_LABEL_	COL1
1	Trt	Treatment	1.00
1	NObs	N Obs	161.00
1	WBC_LCLM	Lower 95% CL for Mean	437.93
1	WBC_UCLM	Upper 95% CL for Mean	525.39
1	WBC_Range	Range	974.00
1	WBC_CSS	Corrected SS	12631956.21
1	WBC_Skew	Skewness	0.17
1	WBC_CV	Coeff of Variation	58.34
1	WBC_StdDev	Std Dev	280.98
1	WBC_Kurt	Kurtosis	-1.11
1	WBC_StdErr	Std Error	22.14

Use SQL on these highly normalized tables to compute N(%), union the the tables , order the output and assign all descriptions(using formats)

Finally, transpose the SQL output and do a dumb print using 'proc report'

2. When possible use numeric values for variables. For example '0/1' for 'Male/Female' provides order and bolean information.

3. Choose variable names that can be transposed into useful data. For example, applying 'proc means' to variable 'Wgt' yields 'Wgt_Mean', Wgt_NMiss' ... Wgt_Probt'. For nominal(discrete level data), ie variable Sex, create formats like

```
Value  Sex2Lvl
```

```
            0='Male'
            1='Female'

   Where Sex is the variable name and in the raw input data.
   In addition, if you want ‘Male’ first then assign 0 to sex.


4. Use formats for page and line order, line selection and line
   description. For example the third major grouping for page 2
   would be 'White Blood Cell Count' with minor groupings of
   'N' and 'Mean(DS)' in that order. Only variables with 'X' would
   appear in the report.

   value $Mjr2Des

  'Wbc' = '020300 White Blood Cell Count'

   value $Mnr2Des

   'N'       =  '01 X N'
   'NMiss'   =  '02   Number Missing'
   'Mean'    =  '03 X Mean(SD)'

5. Create two sets of page numbers, one for Microsoft headers and
   footers and another for the body of the report.

6. For the Total column just replicate the entire data and assign a
   new treatment number.
```

CONTACT INFORMATION

Author Name	Roger DeAngelis
Company	AMGEN
Address	One AMGEN Center Dr.
City state ZIP	Thousand Oaks CA 91320
Work Phone:	805-447-9821
Cell:	845-430-1533
Email:	xlr82sas@hotmail.com
Web:	