

PROGRAMMING FOR PROC TABULATE

DANIEL MORRIS,
AMADEUS SOFTWARE TRAINING,
OXFORD, U.K.

ABSTRACT

This paper describes the preparation of data and the construction of SAS® data sets for use with Proc Tabulate. Two SAS procedures are discussed: Proc Freq and Proc Summary, and their use in preparing data. Data step techniques are outlined and examples of tables created using these techniques are presented. Percentage tables are discussed and some tips on denominator specifications are given.

INTRODUCTION

There are several reasons for the preparation of data:

1. to make efficient use of SAS procedures
2. to overcome limitations with Proc Tabulate
3. to provide a better data structure for Proc Tabulate

We discuss the first 2 aspects in this paper.

1. EFFICIENT USE OF SAS PROCEDURES

Proc Freq and Proc Summary can be used to perform calculations on the data. Both of these procedures produce an output SAS data set that Proc Tabulate can use.

Some demographic data is used for a series of examples. The data consists of:

PATIENT: Patient number
SEX: Male or Female
TREATMNT: DRUG A or DRUG B
AGE: Patients age
HEIGHT: Patients height
WEIGHT: Patients weight

In this example, Proc Freq is used to provide counts and percentages:

```
PROGRAM EDITOR
Command ==>

00001 proc freq data=rwdata.demogl;
00002 table sex*treatmnt/noprint out=stats;
00003 run;
00004
00005 proc print data=stats;
00006 run;
00007
00008 proc tabulate data=stats;
00009 class sex treatmnt;
00010 var count;
00011 table sex,treatmnt*count='''sum='''*f=7.
00012 /rts=16 box='Using Proc Freq Output - Counts';
00013 run;
00014
00015 proc format;
00016 picture perc low-high='000%';
00017 run;
00018
00019 proc tabulate data=stats;
00020 class sex treatmnt;
00021 var percent;
00022 table sex,treatmnt*percent='''sum='''*f=perc7.
00023 /rts=16 box='Using Proc Freq Output - Percentages';
00024 run;
```

Counts and percentages are easily created in a SAS data set by Proc Freq and then tabled.

OUTPUT
Command ==>

OBS	SEX	TREATMNT	COUNT	PERCENT
1	Female	DRUG A	21	30.0000
2	Female	DRUG B	12	17.1429
3	Male	DRUG A	18	25.7143
4	Male	DRUG B	19	27.1429

Using Proc Freq Output - Counts	Treatment which Patient received	
	DRUG A	DRUG B
Sex of patient		
Female	21	12
Male	18	19

Using Proc Freq Output - Percentages	Treatment which Patient received	
	DRUG A	DRUG B
Sex of patient		
Female	30%	17%
Male	25%	27%

Summary data sets can be created and then tabled. Here we round the values of the average weight so that the truncation of the picture does not give misleading results.

Notice the similarity between the Proc Summary Class and Var statements and the Proc Tabulate Class and Var statements.

```
PROGRAM EDITOR
Command ==>

00001 proc summary data=rwdata.demogl nway;
00002 class sex treatmnt;
00003 var weight;
00004 output out=stats mean=avgwt;
00005 run;
00006
00007 proc print data=stats( drop=_type_ _freq_);
00008 run;
00009
00010 data stats;
00011 set stats;
00012 avgwt=round(avgwt,.01);
00013 run;
00014
00015 proc format;
00016 picture kilos low-high='0000.00 kgs';
00017 run;
00018
00019 proc tabulate data=stats;
00020 class sex treatmnt;
00021 var avgwt;
00022 table sex,treatmnt*avgwt='''sum='''*f=kilos.
00023 /rts=18 box='Using Proc Summary Output - Average Weights';
00024 run;
```

Proc Summary can create an aggregate data set containing many different statistics on different levels in the data. Often, the creation of a summary data set for all possible aggregations is done first. Tabling can then be performed on the summary data set rather than the original data.

OUTPUT
Command ==>

OBS	SEX	TREATMNT	AVGWT
1	Female	DRUG A	66.4286
2	Female	DRUG B	85.2917
3	Male	DRUG A	89.6889
4	Male	DRUG B	87.3158

Using Proc Summary Output - Average Weights	Treatment which Patient received	
	DRUG A	DRUG B
Sex of patient		
Female	66.43 kgs	85.29 kgs
Male	89.69 kgs	87.32 kgs

2. OVERCOMING LIMITATIONS

Although an excellent procedure, Proc Tabulate does have a few limitations. For example, the inclusion of percentiles and medians in tables is not available from Tabulate directly. Also the addition of blank lines and removal of selected separator lines is not possible.

Circumventions to some limitations are provided by the capabilities of the data step in a data preparation stage.

INCORPORATING MEDIANS

In this example, medians are calculated from Proc Univariate and included in the table. This technique can be extended to include any percentiles, allowing upper and lower quartile ranges to be shown.

PROGRAM EDITOR
Command ==>

```

00001 proc summary data=rwdata.demogl nway;
00002   class sex treatmnt;
00003   var weight;
00004   output out=stats mean=avgwt;
00005 run;
00006
00007 proc print data=stats( drop=_type_ _freq_);
00008 run;
00009
00010 data stats;
00011   set stats;
00012   avgwt=round(avgwt,.01);
00013 run;
00014
00015 proc format;
00016   picture kilos low-high='0000.00 kgs';
00017 run;
00018
continued on next page.....

```

PROGRAM EDITOR
Command ==>

```

continued from previous page.....
00019 proc sort data=rwdata.demogl out=demsrt;
00020   by sex treatmnt;
00021 run;
00022
00023 proc univariate data=demsrt noprint;
00024   by sex treatmnt;
00025   var weight;
00026   output out=medians pctlpts=50 pctlpre=medwt;
00027 run;
00028
00029 proc print data=medians;
00030 run;
00031
00032 data allstats;
00033   merge stats medians;
00034   by sex treatmnt;
00035 run;
00036
00037 proc tabulate data=allstats;
00038   class sex treatmnt;
00039   var avgwt medwt50;
00040   table sex,treatmnt*
00041     (avgwt='Average Weight' medwt50='Median Weight')
00042     *sum='*f=kilos.
00043     /rts=18 box='Using Medians from Proc Univariate';
00044 run;

```

OUTPUT
Command ==>

	OBS	SEX	TREATMNT	AVGWT
proc summary	1	Female	DRUG A	66.4286
data set:	2	Female	DRUG B	85.2917
	3	Male	DRUG A	89.6889
	4	Male	DRUG B	87.3158

	OBS	SEX	TREATMNT	MEDWT50
proc univariate	1	Female	DRUG A	62.50
data set:	2	Female	DRUG B	89.00
	3	Male	DRUG A	88.75
	4	Male	DRUG B	86.00

final table:

Using Medians from Proc Univariate	Treatment which Patient received			
	DRUG A		DRUG B	
	Average Weight	Median Weight	Average Weight	Median Weight
Sex of patient				
Female	66.43 kgs	62.50 kgs	85.29 kgs	89.00 kgs
Male	89.69 kgs	88.75 kgs	87.32 kgs	86.00 kgs

ADDING BLANK LINES TO A TABLE

Adding some blank lines to a table can be achieved by using the data step:

```
PROGRAM EDITOR
Command ==>

00001 data blines;
00002   set allstats end=e;
00003   output;
00004   if e then do;
00005     sex=''; treatmnt=''; avgwt=.; medwt50=.;
00006     output;
00007   end;
00008 run;
00009
00010 options missing=' ';
00011
00012 proc format;
00013   value $blanks ' ','DRUG A'='DRUG A';
00014 run;
00015
00016 proc tabulate data=blines noseps missing;
00017   class sex treatmnt;
00018   var avgwt medwt50;
00019   table sex,treatmnt*
00020     (avgwt='Average Weight' medwt50='Median Weight' )
00021     *sum='**f=kilos.
00022     /rts=18 box='Inserting blank lines in the table ';
00023   format treatmnt $blanks.;
00024 run;
```

OUTPUT
Command ==>

Inserting blank lines in the table	Treatment which Patient received			
	DRUG A		DRUG B	
	Average Weight	Median Weight	Average Weight	Median Weight
Sex of patient				
Female	66.43 kgs	62.50 kgs	85.29 kgs	89.00 kgs
Male	89.69 kgs	88.75 kgs	87.32 kgs	86.00 kgs

Changing the order of the rows is accomplished by some data manipulation. This technique allows you to put the data in any order, not simply sorted, data or formatted.

```
PROGRAM EDITOR
Command ==>

00001 data blines;
00002   set allstats end=e;
00003   output;
00004   if e then do;
00005     sex=''; treatmnt=''; avgwt=.; medwt50=.;
00006     output;
00007   end;
00008 run;
00009
00010 data males females blanks;
00011   set blines;
00012   if sex='Male' then output males;
00013   else if sex='Female' then output females;
00014   else if sex=' ' then output blanks;
00015 run;
00016
00017 data backagin;
00018   set males blanks females;
00019 run;
00020
00021 options missing=' ';
00022
00023 proc format;
00024   value $blanks ' ','DRUG A'='DRUG A';
00025 run;
00026
00027 proc tabulate data=backagin noseps missing order=data;
00028   class sex treatmnt;
00029   var avgwt medwt50;
00030   table sex='',treatmnt*
00031     (avgwt='Average Weight' medwt50='Median Weight' )
00032     *sum='**f=kilos.
00033     /rts=18 box='Reordering the rows with the data step';
00034   format treatmnt $blanks.;
00035 run;
```

OUTPUT
Command ==>

Reordering the rows with the data step	Treatment which Patient received			
	DRUG A		DRUG B	
	Average Weight	Median Weight	Average Weight	Median Weight
Male	89.69 kgs	88.75 kgs	87.32 kgs	86.00 kgs
Female	66.43 kgs	62.50 kgs	85.29 kgs	89.00 kgs

PERCENTAGES

Creating tables of percentages with unusual denominators is quite easy given the following procedure:

1. From a sketch of your table, draw a box around the numbers you want to add up to 100.
2. Identify the variables whose values change in the box.

Example 1:

OUTPUT
Command ==>

	Female		Male		Total	
	Treatment which Patient received		Treatment which Patient received		Treatment which Patient received	
	DRUG A	DRUG B	DRUG A	DRUG B	DRUG A	DRUG B
	PCTN	PCTN	PCTN	PCTN	PCTN	PCTN
Age in years at start of Trial						
20 - 29			6		3	
30 - 39	19	8	28	16	23	13
40 - 49	14	33	22	42	18	39
50 - 59	48	42	28	25	38	32
60 - 69	19	17	17	15	18	16
total=100						

AGE changes in the box.

PROGRAM EDITOR

Command ==>

```
00001 proc format;
00002     value ageband
00003         low-19='Under 20'
00004         20-29='20 - 29'
00005         30-39='30 - 39'
00006         40-49='40 - 49'
00007         50-59='50 - 59'
00008         60-69='60 - 69'
00009         70-high='70 and over';
00010 run;
00011
00012 proc tabulate data=rwdata.demogl format=7.;
00013     class sex treatmnt age;
00014     table age,sex=' '*treatmnt*pctn<age>
00015         all='Total'*treatmnt*pctn<age>;
00016     format age ageband.;
00017 run;
```

Example 2:

OUTPUT

Command ==>

	Female		Male		Total	
	DRUG A	DRUG B	DRUG A	DRUG B	DRUG A	DRUG B
Age in years at start of Trial						
20 - 29			3		1	
30 - 39	12	3	14	8	13	6
40 - 49	9	12	11	22	10	17
50 - 59	30	15	14	14	21	14
60 - 69	12	5	8	8	10	7

total=100

Age and Treatmnt change their values in the box:

PROGRAM EDITOR

Command ==>

```
00001 proc tabulate data=rwdata.demogl format=7.;
00002     class sex treatmnt age;
00003     table age,sex=' '*treatmnt=' '*pctn<age*treatmnt>'
00004         all='Total'*treatmnt=' '*pctn<age*treatmnt>'
00005     format age ageband.;
00006 run;
```

Example 3:

PROGRAM EDITOR

Command ==>

```
00001 proc tabulate data=rwdata.demogl format=7.;
00002     class sex treatmnt age;
00003     table age,sex=' '*treatmnt=' '*pctn<sex*treatmnt>'
00004         all='Total'*treatmnt=' '*pctn<treatmnt>'
00005     format age ageband.;
00006 run;
```

Sex and Treatmnt change their values in the first box, treatmnt changes its value in the 2nd:

OUTPUT

Command ==>

	Female		Male		Total	
	DRUG A	DRUG B	DRUG A	DRUG B	DRUG A	DRUG B
Age in years at start of Trial						
20 - 29			100		100	
30 - 39	31	8	35	23	69	31
40 - 49	16	21	21	42	37	63
50 - 59	40	20	20	20	60	40
60 - 69	33	17	25	25	58	42

CONCLUSION

Proc Tabulate is an excellent procedure for the production of tables, and is greatly enhanced by the ability of other procedures such as Proc Freq and Proc Summary to feed through statistics and counts. Some limitations of the procedure can be addressed through data manipulation techniques and a detailed knowledge of the data step mechanics helps enormously. Other aspects of the SAS System, such as formats, are very helpful in creating the desired report.

The paper has also touched on the subject of denominator definitions for percentages and offered an alternative method for determining the correct denominator definition easily.

Authors address:

Daniel Morris,
Amadeus Software Training,
34 High Street,
Woodstock,
Oxfordshire, OX20 1TG,
United Kingdom.
(44) 0993 813188

Trademarks:

SAS is a registered trademark of SAS Institute, Cary, NC, USA.