Paper 1594-2014

GENERATING DYNAMIC TABLES USING PROC SQL AND PROC TABULATE

Lovedeep Gondara, BC Cancer Agency

ABSTRACT

PROC TABULATE is the most widely used reporting tool in SAS®, along with PROC REPORT. Any kind of report with the desired statistics can be produced by PROC TABULATE or PROC REPORT. In clinical studies summary table commonly known as Table-1 is produced for summary statistics for various variables of interest. Comparison between different treatment groups can also be reported using P-values from different statistical tests. This is usually done by running different Procedures depending on type of test required and type of variable used and then grouping the output manually in a word processor software which can be a tedious task especially if your dataset changes over time or if it is a periodical report. In here we introduce a macro that is capable of performing all major statistical tests and then combining their output in a single dataset that can be output as a formatted summary report using PROC TABULATE or PROC REPORT.

INTRODUCTION

A summary often called as Table-1 is the very first table presented in published articles and is mainly used to describe a cohort or to examine crude relation or associations between different variables and the variable of interest, mainly the outcome variable. During the process of building such type of tables, continuous changes or refinements are often encountered which result in running the summary statistics again and then populating the formatted output report.

In here we present a macro that is capable of performing all these tasks automatically, it primarily focuses on a two level outcome variable such as comparison between Treatment and non treatment group by exploring different variables. This work is based on twoway macro presented by Lan Hu[1] and Xiaoqiang Wang[1].
A typical summary table in publication's looks likes

	Active (N=31)	Placebo (N=31)	Overall (N=60)	P-value
Age (years)				*0.9528*
N	31	29	60	
Mean	51.4	50.1	50.8	
Standard Deviation	13.23	13.22	13.13	
Minimum	32.0	23.0	23.0	
Maximum	75.0	77.0	77.0	
Gender				*0.2681*
Male	22(71.0%)	16(57.1%)	38(64.4%)	
Female	9(29.0%)	12(42.9%)	21(35.6%)	
Race				*0.9270*
White	18(58.1%)	18(62.1%)	36(60.0%)	
Black	10(32.3%)	8(27.6%)	18(30.0%)	
Other*	3(9.7%)	3(10.3%)	6(10.0%)	

Table1: Desired Output
In general to produce a similar summary table we need to run different procedures to get P-values and for specific type of test requested and required.

SUMMARY TABLE MACRO

Macro call –

```
 %summary_table(
dsn=sgf2014_test,
xvars=age.wilcoxon ethnicity.fisher smoke_ever.fisher weight.ttest,
yvar = class,
comp=1,
out_type=1,
path=
)
```

Where –

Dsn - Dataset name
Xvars- Variables to summarize followed by a keyword for test after period if
P-values are desired.
Yvar - variable to base summary on its different categories
comp -1 if want p-values in output , 0 – just summary
Out_type – 1 – Output created using PROC Tabulate, 2 – Output created using PROC Report
Path – Path for output RTF file

WORKING

1. To start a configuration dataset is created that stores the keywords used for various tests called in macro.

```
DATA config;
        INFILE DATALINES4 DSD DLM=',';
        INPUT freq_output_option:$8.
freq_table_option:$10.
freq_exact_statement:$20. test:$50.;
DATALINES4;
AJCHI,CHISQ,,Continuity-adjusted chi-
square for 2x2 tables
CMHCOR,CMH2,,Cochran-Mantel-Haenszel
correlation statistic
CMHGA,CMH,,Cochran-Mantel-Haenszel general
association statistic
CMHRMS,CMH2,,Cochran-Mantel-Haenszel row
mean scores (ANOVA) statistic
CONTGY,CHISQ,,Contigency coefficient
CRAMV,CHISQ,,Cramer%'s V
FISHER,FISHER,,Fisher's exact test
GAMMA,MEASURES,,gamma
JT,JT,EXACT JT;,Jonckheer-
.....
```

freq_output_option	freq_table_option	freq_exact_statement	test
AJCHI	CHISQ		Continuity-adjusted chi-square for 2x2 tables
CMHCOR	CMH2		Cochran-Mantel-Haenszel correlation statistic
CMHGA	CMH		Cochran-Mantel-Haenszel general association statis
CMHRMS	CMH2		Cochran-Mantel-Haenszel row mean scores (ANOVA) st
CONTGY	CHISQ		Contigency coefficient
CRAMV	CHISQ		Cramer%'s V
FISHER	FISHER		Fisher's exact test
GAMMA	MEASURES		gamma
JT	JT	EXACT JT;	Jonckheer-Terpstra test
KENTB	MEASURES		Kendall's tau-b
LAMBDAS	MEASURES		lambda symmetric
LAMCR	MEASURES		lambda asymmetric (C|R)

Table2: Configuration file

2. Input variables are scanned to separate Statistical test keywords from input variables.

```
%let i=1;
%let xvar = %scan(&xvars, &i, %str( ));
%do %while(&xvar ne);
        %put PROCESSING variable &xvar;
        %let j=1;
        %let part = %scan(&xvar, &j, .);
        %do %while(&part ne %str());
                %let part&j = &part;
                %let j = %eval(&j+1);
                %let part = %scan(&xvar, &j, .);
        %end;
        %let j = %eval(&j-1);
```

```
        %let itemize=0;
        %let keyword=;
        %let exact=0;
       %let transpose_freq_test=0;


        %if &j=2 %then %do;

                        %let xvar = &part1;
                        %let keyword = &part2;

        %end;
```

3. For categorical variables counts and percentages are generated using PROC FREQ and if any statistical test are requested then the keyword is used to declare it.

```
PROC FREQ DATA=&tmpdsn6 ORDER=DATA NOPRINT;
     TABLE _1 / OUT=&tmpdsn5(KEEP=count percent)  %if &missing %then MISSING;;
     WEIGHT _freq_ %if &zeros %then /ZEROS;;
     WHERE _type_=2;

……..More code……..

%if %sysfunc(inputc(&keyword, &freq_table_option_fmt)) eq %str() %then %let keyword =
&cat_test;
     PROC FREQ DATA=&tmpdsn3 ORDER=DATA NOPRINT;
     TABLE %if &transpose_freq_test %then &yvar*&xvar; %else &xvar*&yvar; /
%sysfunc(inputc(&keyword, &freq_table_option_fmt));
     %if &exact %then %sysfunc(inputc(&keyword, &freq_exact_statement_fmt));
      WEIGHT _freq_;
      OUTPUT OUT=&tmpdsn2 &keyword;
```

4. If it is a continuous variable then PROC MEANS and PROC NPAR1WAY are used for required statistics.

```
PROC MEANS DATA=&dsn(KEEP=&xvar &yvar %if %sysevalf(%superq(weight) ne, boolean) %then
&weight; RENAME=(&yvar=_2 &xvar=_1)) &means_statistics
     %if &zeros %then COMPLETETYPES; %if &missing %then MISSING; NOPRINT;
     CLASS _2/%if &zeros and &preloadable_Y %then PRELOADFMT; %if &mlf %then MLF;;
                       VAR _1;
     %if %sysevalf(%superq(weight) ne, boolean) %then %str(FREQ &weight;);
     OUTPUT OUT=&tmpdsn1(KEEP=_2 _type_ &means_statistics)
     %do j=1 %to &N_MEANS_STATISTICS;
     &&orig_means_statistics&j = &&orig_means_statistics&j
     %end;
         ;

%if not %sysfunc(inputn(&keyword, &npar1way_option_fmt)) %then %let keyword=&con_test;

PROC NPAR1WAY DATA=&dsn(KEEP=&xvar &yvar) &keyword NOPRINT %if &missing %then
MISSING;;
     CLASS &yvar;
     VAR &xvar;
     %if &exact eq e or &exact eq E %then EXACT;;
     %if %sysevalf(%superq(weight) ne, boolean) %then %str(FREQ &weight;);
     OUTPUT OUT=&tmpdsn2 &keyword;
```

5. Then output from both is combined into a single raw data file which has variable names, counts, percentages, summary statistics for continous variables, Test statistics and P values.

6. Raw output table is broken down for categorical and continous variables for proper formatting required for output.

For continous

```
data var_table;
set output;
if categorical=0;
run;
data var_table;
set var_table;
xcat=_name_;
class1=input(round(_stat_1, 0.01), $12.);
class2=input(round(_stat_2,0.01), $12.);
total=input(round(_stat_,0.01), $12.);
run;
data var_table;
set var_table;
dsid=open('var_table');
if varnum(dsid,'pt2_wil')=0 then
pt2_wil=.;
if varnum(dsid,'p_f')=0 then p_f=.;
if pt2_wil ne . then
p_v=input(round(pt2_wil,0.0000001), $21.);
if p_f ne . then
p_v=input(round(p_f,0.0000001), $21.);
run;
```

For categorical

```
data cat_table;
set output;
if categorical=1;
run;
data cat_table;
set cat_table;
c1=put(COUNT_1, $9.);

p1=put(pct_col_1, $9.);
class1=
trim(c1)||'('||strip(p1)||'%'||')';


c2=put(COUNT_2, $9.);

p2=put(pct_col_2, $9.);
class2=
trim(c2)||'('||strip(p2)||'%'||')';

c3=put(COUNT, $9.);

p3=put(percent, $9.);
total= trim(c3)||'('||strip(p3)||'%'||')';
if p_pchi ne . then
p_v=input(round(p_pchi,0.0000001), $21.);
if xp2_fish ne . then
p_v=input(round(xp2_fish,0.0000001),
$21.);

run;
```

7. Now as in same format and having similar variable names, both of above tables can be appended.

```
data var_table ;
set var_table(keep=varname xcat class1 class2 total p_v categorical) ;
run;
data cat_table ;
set cat_table(keep=varname xcat class1 class2 total p_v categorical) ;
run;
proc append data=var_table base=cat_table;
run;
```

8. After making some other minor changes resulting dataset is output, which looks like

varname	categorical	xcat	class1	class2	total	p_v
age_c	1	<35	300(12%)	33(30%)	333(12%)	.0001
age_c	1	35-49	1251(48%)	53(48%)	1304(48%)	
age_c	1	>=50	1033(40%)	25(23%)	1058(39%)	
ethnicity_new	1		258(10%)	17(15%)	275(10%)	0.5539558
ethnicity_new	1	Aboriginal	12(0%)	1(1%)	13(0%)	
ethnicity_new	1	Chinese	227(9%)	6(5%)	233(9%)	
ethnicity_new	1	European	1815(70%)	75(68%)	1890(70%)	
ethnicity_new	1	Other	137(5%)	8(7%)	145(5%)	
ethnicity_new	1	Other Asia	135(5%)	4(4%)	139(5%)	
l_edu_level	1		262(10%)	19(17%)	281(10%)	0.8215661
l_edu_level	1	High School or Less	397(15%)	17(15%)	414(15%)	
l_edu_level	1	Trade school or College	687(27%)	29(26%)	716(27%)	
l_edu_level	1	Some or any University	1238(48%)	46(41%)	1284(48%)	
l_partners	1		282(11%)	18(16%)	310(12%)	0.001131
l_partners	1	0-5	1305(51%)	35(32%)	1340(50%)	
l_partners	1	6-10	533(21%)	32(29%)	565(21%)	
l_partners	1	>10	454(18%)	26(23%)	480(18%)	
smoke_ever_n	1		314(12%)	19(17%)	333(12%)	0.0465775
smoke_ever_n	1	No	1394(54%)	47(42%)	1441(53%)	
smoke_ever_n	1	Yes	876(34%)	45(41%)	921(34%)	

Fig 1: Output dataset

This above resulting dataset can be output using PROC TABULATE or PROC REPORT as desired.

EXAMPLES –

Ex-1: Using combination of continuous and categorical variables just to create a simple summary report without p-values.

Sample Dataset –

```
Proc format;
value $gender
'0'='Female'
'1'='Male';
value $Race
'1'='European'
……
data sgf2014;
input subjid gender $ trt race $ age @@;
label gender = "Gender"
race = "Race"
age = "Age (years)";
format race $race. gender $gender.;
datalines;
101 0 1 3 37 301 0 1 1 70 501 0 1 2 33 601 0 1 1 50 701 1 1 1 60
102 1 2 1 65 302 0 1 2 55 502 1 2 1 44 602 0 2 2 30 702 0 1 1 28
……..;
```

Here gender and race are categorical variables where age is a continous variable and trt is treatment group on basis of which our summary table will be created.

Macro call-

```
%summary_table(
dsn=sgf2014,
xvars=age Race Gender ,
yvar = trt,
comp=0,
out_type=1,
path='C:\Documents and Settings\Lovedeep.Gondara\sgf\output1.rtf'
)
```

Using PROC TABULATE

Variable Name		Treat1	Treat2	Total	
Gender	Female	16(42%)	12(57%)	28(47%)	1
	Male	22(58%)	9(43%)	31(53%)	1
Race	European	19(50%)	16(76%)	35(59%)	1
	Asian	13(34%)	5(24%)	18(31%)	1
	American	6(16%)	0(0%)	6(10%)	1
age	N	38	21	59	1
	MEAN	49.76	52.62	50.78	1
	MEDIAN	46	53	49	1
	STD	12.95	13.88	13.24	1

Using PROC REPORT

Variable	Category	Treat1	Treat2	Total
Gender	Female	16(42%)	12(57%)	28(47%)
	Male	22(58%)	9(43%)	31(53%)
Race	European	19(50%)	16(76%)	35(59%)
	Asian	13(34%)	5(24%)	18(31%)
	American	6(16%)	0(0%)	6(10%)
age	N	38	21	59
	MEAN	49.76	52.62	50.78
	MEDIAN	46	53	49
	STD	12.95	13.88	13.24

Table3: Result

With P-values

```
 %summary_table(
dsn=sgf2014,
xvars=age.wilcoxon Race Gender ,
yvar = trt,
comp=1,
out_type=1,
path='C:\Documents and Settings\Lovedeep.Gondara\sgf\output3.rtf`
)
```

Using PROC TABULATE

Variable Name		Treat1	Treat2	Total	P Value	
Gender	Female	16(42%)	12(57%)	28(47%)	0.2680754	1
	Male	22(58%)	9(43%)	31(53%)		1
Race	European	19(50%)	16(76%)	35(59%)	0.0685859	1
	Asian	13(34%)	5(24%)	18(31%)		1
	American	6(16%)	0(0%)	6(10%)		1
age	N	38	21	59	0.3699349	1
	MEAN	49.76	52.62	50.78		1
	MEDIAN	46	53	49		1
	STD	12.95	13.88	13.24		1

Using PROC REPORT

Variable	Category	Treat1	Treat2	Total	P-Value
Gender	Female	16(42%)	12(57%)	28(47%)	0.2680754
	Male	22(58%)	9(43%)	31(53%)	
Race	European	19(50%)	16(76%)	35(59%)	0.0685859
	Asian	13(34%)	5(24%)	18(31%)	
	American	6(16%)	0(0%)	6(10%)	
age	N	38	21	59	0.3699349
	MEAN	49.76	52.62	50.78	
	MEDIAN	46	53	49	
	STD	12.95	13.88	13.24	

Table4: results

CONCLUSION

Using this macro users can easily generate almost all required summary statistics for a basic summary table and can request p-values to compare groups for y variable.
A nicely formatted output dataset is created which contains all the information and can be used if custom formatted report other than available in macro is required. As use of PROC TABULATE generates an extra column containing counts, manual edition will be required. To overcome this and make macro truly autonomous, user can request output via PROC REPORT.

FUTURE WORK

Future work is required on this macro in order for it to be used for y-variable having more than two levels, further work is also required to make this macro less computational intensive as while calculating exact values it can use large amount of system resources.

REFERENCES

- Lan Hu, Xiaoqiang Wang, *The Ultimate Macro to Generate Summary Tables.*
- Kathryn Martin, *A Researcher's Guide to Making Descriptive and Analytic Tables that are Ready to Publish, WUSS 2006*
- Zang Xiong, *%YAMGAST: Yet Another Macro to Generate a Summary Table, PharmaSUG 2008*
- Jack Shostak, *SAS Programming in the Pharmaceutical Industry, SAS Press, August 2005.*
- Yinmei Zhou, Lijun Zhang, Michael L. Hancock, *%SummaryTable: A SAS Macro to Produce a Summary Table in Clinical Trial, PharmaSUG 2006*
- Jun Zuo, Carl R. Haske, *Creating Clinical Trial Summary Table Containing P-Values: A Practical Approach Using Standard SAS Macros, SUGI 22*

CONTACT INFORMATION

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

Name: Lovedeep Gondara
Organization: BC Cancer Agency
Address: 686 west broadway
City, State ZIP: Vancouver BC
Email: Lovedeep.gondara@bccancer.bc.ca