

%TABLEMAKER: Data Reporting Made Easy

Bernadette H. Johnson, Premier, Inc, Charlotte, NC

ABSTRACT

Reporting data can be challenging, especially when analyzing both continuous and categorical data with the goal of presenting all of the results in the same table. In three steps, the TABLEMAKER macro generates user-selected descriptive statistics and presents results as an integrated table. In Step 1, the user provides overall table parameters such as: page orientation and data set name. In Step 2, the SAS variables and the desired list of descriptive statistics are defined for each horizontal table section. Step 3 generates and executes the code to create the table. The TABLEMAKER macro is a user-written tool that provides an easy way to create tables for analysts, statisticians, researchers.

INTRODUCTION

Creating tables typically involves two steps: first, writing programming code to analyze the data and second, writing additional code to report the results in a customized table. Categorical data – data having discrete values that can be easily counted (i.e. gender) – are often reported showing the frequency and proportions for each unique value. Continuous data – non-categorical data (i.e. height) – are summarized to report the minimum, maximum, range, mean, standard deviation, median and more. The TABLEMAKER macro performs the analysis for both types of data and integrates the results into report-ready tables.

FREQUENCY TABLE

A simple frequency distribution table of the categorical variable GENDER is shown in Table 1.

Table 1: Frequency of Gender		
Gender	F	197(58.6%)
	M	139(41.4%)

The code needed to analyze and generate the table is:

- ❶ SETUP Specify reporting dataset


```
%TABLEmaker (action=SETUP
, data=demog);
```
- ❷ DEFINE Add table section type, analysis variable name


```
%TABLEmaker( action=DEFINE
, var=gender
, type=FREQ
, label=Gender);
```
- ❸ CREATE Assign title and generate table


```
%TABLEmaker(action=CREATE
, title1=Table 1: Frequency of Gender);
```

DESCRIPTIVE STATISTICS TABLE

A summary table of the continuous variable CHARGES, hospital billing charges, is shown in Table 2.

Table 2: Total Hospital Charges		
Total Charges (\$)	N=	336
	Minimum-Maximum	20,047.9 - 285,950.5
	Mean(Standard Deviation)	47,024.1(±38,056.3)
	Median	34,206.1
	IQR	24,687.8 - 52,016.5

The code needed to analyze and generate the table is:

- 1 SETUP Specify reporting dataset

```
%TABLEmaker(action=SETUP  
, data=demog);
```
- 2 DEFINE Add table section type, analysis variable name and list of statistics in desired reporting order

```
%TABLEmaker(action=DEFINE  
, var=total_charges  
, type=STAT  
, stat=N MINMAX MEANSTD MEDIAN IQR  
, label= Total Charges ($));
```
- 3 CREATE Assign title and generate table

```
%TABLEmaker(action=CREATE  
, title1=Table 2: Total Charges);
```

The macro submits the following PROC UNIVARIATE code to generate the descriptive statistics:

```
proc univariate data=__TMPRPT noprint;  
var total_charges;  
output out=__TMPRPT2  
n=n min=min max=max median=median std=std mean=mean q1=q1 q3=q3;  
run;
```

COMBINING TABLE SECTIONS

Table 3 illustrates the flexibility and ease of creating more complex tables by defining the analysis and reporting parameters using the TABLEMAKER macro. Larger tables are created by combining multiple table definitions. When used individually, each definition yields a table. When combined, each definition is used to describe a horizontal report section within the larger table.

For each report section, the user options include:

- Ordering of the report sections -- the report order follows the order of the DEFINE statements
- By Group processing to create cross tabulation or two-way tables
- Aggregating values into groups prior to analysis by formatting the analysis variable
- Assigning custom labels for each report section
- Assigning formats for output results
- Limiting which report outcome values are printed by applying where clause processing after the analysis
- Selecting and ordering the descriptive statistics to perform and report. The statistics are reported in the order specified in the DEFINE statement

Table 3 incorporates many of the options to illustrate the level of customization available to the user.

		Age ①				
		Under 15	15 to 30 years old	31 to 60 years old	61 to 90 years old	Over 90 years old
Patient Count	N=	17	24	125	163	7
Gender	M ②	11(64.7%)	7(29.2%)	53(42.4%)	68(41.7%)	
Race ③	White	12(70.6%)	14(58.3%)	90(72.0%)	124(76.1%)	6(85.7%)
	Other	5(29.4%)	10(41.7%)	35(28.0%)	39(23.9%)	1(14.3%)
Frequency Distribution of Length of Stay	Outpatient Visit ④	1(5.9%)	3(12.5%)	12(9.6%)	9(5.5%)	
	>= 1 Week	8(47.1%)	17(70.8%)	96(76.8%)	102(62.6%)	5(71.4%)
	> 1 Week	8(47.1%)	4(16.7%)	17(13.6%)	52(31.9%)	2(28.6%)
Length of Stay (Inpatients Only) ⑤ (days)	N=	16	21	113	154	7
	Minimum	2	1	1	1	2
	Maximum	63	20	21	30	17
	Median	8	4	4	5	5
	6					
Total Charges (\$)	N=	17	24	125	163	7
	Minimum-Maximum	\$20,048 - \$190,326	\$20,064 - \$85,400	\$20,111 - \$234,553	\$20,134 - \$285,950	\$20,049 - \$43,815
	Mean (Std Dev)	\$67,408 (±\$52,104)	\$34,345 (±\$16,288)	\$45,592 (±\$38,141)	\$48,606 (±\$38,577)	\$29,724 (±\$8,976)
	Median	\$38,346	\$28,952	\$30,567	\$38,388	\$27,948
	IQR ⑦	\$26,496 - \$110,175	\$24,586 - \$36,745	\$24,356 - \$49,469	\$25,850 - \$57,446	\$23,201 - \$40,310

① Create a two-way table by using a BY variable and aggregate values into ranges by assigned a format to the BY variable AGE.

%TABLEmaker(..., by=age, byformat=agegrp., ...);

② Report selected values from entire population. In this example, determine the counts of both males and females, but only report the count and percentage of males in the population

%TABLEmaker(..., rptvalues="M",...);

③ Aggregate values into ranges prior to the analysis by assigning a format to the analysis variable RACE

%TABLEmaker(..., var=race, varformat=\$race., ...);

④ Aggregate continuous variable (LENGTH_OF_STAY) into groups for counts and reporting

%TABLEmaker(..., type=FREQ, varformat=los., ...);

⑤ Use where clause processing to subset population to only analyze data where I_O_FLAG="I"

%TABLEmaker(..., where=i_o_flag='I',...);

⑥ Select and order the descriptive statistics to include in the table

%TABLEmaker(type=STAT, stat=N MIN MAX MEDIAN,...);

⑦ Format the output results to include the dollar sign (\$)

%TABLEmaker(..., outfmt=dollar10., ...);

THE TABLEMAKER MACRO

The TABLEMAKER macro performs three actions: setup, analysis and reporting. The macro automatically runs the code section related to the action as detailed below.

%TABLEmaker(ACTION =SETUP);

ACTION=SETUP:

- Signals that a new table is being defined
- Defines global macro variable that will apply to the overall table, such as:
 - Input dataset name
 - Output dataset name to hold analysis results (default: WORK.TABLEMAKER)
 - By group variable, format, label text (optional).

ACTION=SETUP can be called multiple times for a single table to change the input dataset name.

%TABLEmaker(ACTION =DEFINE);

ACTION= DEFINE describes each horizontal section of the table.

ACTION= DEFINE assigns:

- the type of analysis FREQ (frequency distribution) or STAT (descriptive statistics)
- analysis variable and associated format
- statistics to include in report (STAT report type only)
- categorical values to include in report (FREQ report type only)
- where clause to apply prior to analysis (optional)

%TABLEmaker(ACTION =CREATE);

ACTION= CREATE generates and submits the PROC Report code to produce the table based on the specifications defined above.

The SETUP ACTION

SETUP is used to initialize values used in the entire table. Each parameter is stored as a global macro variable. Global macro variables retain their values across multiple calls to the macro until intentionally changed. Therefore, the values provided in the SETUP step are available during the execution of the DEFINE and CREATE steps.

```
%macro TABLEmaker
```

```
(action=SETUP
```

```
,data=
```

```
,out=TABLEmaker
```

```
,by=
```

```
,byformat=
```

```
,bylabel=);
```

- Action=SETUP defines the initial values for the entire table
- Dataset containing report variables
- Output dataset to contain table results. Default output dataset is WORK.TABLEMAKER(optional)
- Variable to use in cross tabulations. (optional)
- Format to associate with BY variable. (optional)
- Label to associate with BY variable. (optional)

The SETUP segment of the TABLEMAKER macro is:

```

%if %upcase(%str(&action))=%str(SETUP) %then %do;

  %* set global macro vars;
  %global _rptDATA _rptOUT _rptBY;;

  %let _rptDATA=&data;
  %let _rptOUT =&out;
  %let _rptBY=&by;

  %* create empty dataset to hold results for reporting;
  data &_rptOUT ;
  stop;
  run;

  %goto bottom;
%end; %* end of SETUP;

```

THE DEFINE ACTION

DEFINE specifies the analysis and reporting parameters for each table section. The order of the DEFINE statements determines the order of the horizontal sections of the report. Each statement is run independently to generate analysis results. All results are stored in a single dataset used by the CREATE statement to generate the report.

```

%macro TABLEmaker
(action=DEFINE
, type=FREQ
, var=
, varformat=
, outfmt=
, rptvalues=
, stat=
, where=);

```

- Action=DEFINE specifies the analysis and reporting parameters for each table section
- Table type. Valid types: FREQ for frequency distributions, STAT for descriptive statistics. Default: FREQ
- Variable to use in analysis
- Format to associate with analysis variable during analyses (optional)
- Format to associate with analysis variable during reporting (optional). Defaults: BEST. for numeric variables, \$CHAR. for character variables.
- Variable values to include in output. Valid only when TYPE=FREQ. Example: only report results for males after calculating frequency distributions for both males and females.
- Descriptive statistics to include in report. Valid only when TYPE=STAT. The statistics appear in the report in the order listed in STAT=.
- Valid entries are:

N	Count
MIN	Minimum
MAX	Maximum
MINMAX	Minimum- Maximum
MEDIAN	Median
MEAN	Mean
MEANSTD	Mean (\pm std dev.)
IQR	Interquartile Range
- Where clause used to limit data submitted for analysis (optional)

Table 1 used the following DEFINE statement to generate a frequency distribution table:

```
%TABLEmaker( action=DEFINE , type=FREQ , var=gender, label=Gender);
```

The code generated from the DEFINE section to analyze the data:

```
proc tabulate data=demog out=__TMPRPT missing;  
class gender /order=data ;  
table gender , (n="N"*f=comma. colpctn="%"*f=5.1 ) /printmiss ;  
run;
```

THE CREATE ACTION

CREATE concludes the table definitions and reads the results dataset produced in the CREATE step and generates the table. PROC Report is used to produce the table. Any previously assigned Output Delivery System (ODS) output destinations are honored.

```
%macro TABLEmaker
```

```
(action=CREATE  
,title1=  
,title2=  
);
```

- Action=CREATE reads the results dataset and creates the table
- Title 1 text. (optional)
- Title 2 text. (optional)

CONCLUSION

SAS can be used to create user-friendly reporting tools such as the TABLEMAKER macro. Easy-to-use reporting tools often enable researchers, statisticians, and less experienced programmers to generate report-ready results on their own without requesting the assistance of analysts which can save both time and money. Also, TABLEMAKER produces consistent report layouts for a more unified look in reports.

CONTACT INFORMATION

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

Bernadette H. Johnson
Premier, Inc.
P.O. Box 668800
Charlotte, NC 28266
Phone: (704) 733-2928
E-mail: Bernadette_Johnson@premierinc.com

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.

Other brand and product names are trademarks of their respective companies.