

## Efficiently Produce Descriptive Statistic Summary Tables with SAS Macros

Chunmao Wang, Quintiles, Rockville, MD

### ABSTRACT

In the pharmaceutical industry, two types of tables are commonly used: One is for categorical variables, computing frequency counts and percentages; another is for continuous variables, computing means, standard derivation, medians, min and max, confidence limits, and ranges, etc. Producing these tables is simple yet trivial, and sometime cumbersome and time consuming, as many variables and many conditions might be requested. Hence a template or tool that can help to create these tables is very useful. In this paper we introduce two simple SAS macros that can be incorporated into your SAS programs easily to generate descriptive statistic summary tables. They will not only save a lot of time but also improve the quality.

### INTRODUCTION

In the pharmaceutical industry, most tables are for descriptive statistic summary. Two types of them are commonly used: One is for categorical variables, computing frequency counts and percentages; another is for continuous variables, computing means, standard derivation, medians, min and max, confidence limits, and ranges, etc. Producing these tables is simple yet trivial, and sometime cumbersome and time consuming, as many variables and many conditions might be requested. Hence a template or tool that can help to create these tables is very useful.

In this paper we introduce two simple SAS macros that can be incorporated into your SAS programs easily to generate descriptive statistic summary tables. They will not only save a lot of time but also improve the quality.

### COUNTS FOR CATEGORICAL VARIABLES

If the variables are categorical, we usually need to count for each categorical and compute the percentage. For example:

	Drug	Placebo	Total
Gender			
Male	20 (40.0%)	25 (50.0%)	45 (45.0%)
Female	30 (60.0%)	25 (50.0%)	55 (55.0%)

Usually we can easily use proc freq to compute the counts and percentages. However, if there are many variables, and for each variable there are many categories, and there are many the conditions for columns, table will be very large, and the SAS program will turn to complicate. Here we introduce a simple macro:

```
%macro _freq(var);
  proc freq data=D20 noprint;
    tables ROW&var /
      out=F&var (rename=(ROW&var=ROW) where=(PERCENT>.));
  run;
%mend;
```

This macro is pretty simple; however, it is essential to use it efficiently. Here is an example:

```
data F;
  ROW=.;
  COUNT=.;
  PERCENT=.;
  delete;
run;

data _NULL_;
  do var=1 to 2;
    call symput("var",put(var,Z3.0));
    call execute('%_freq(&var);');
    call execute('
```

```

data F&var;
  set F&var;
  ROW=&var;
run;

proc append base=F data=F&var;
run;
');
end;
run;

```

The first step create an empty dataset which will be filled in the next step. In the second step, we use call execute, so we can run the macro repeatedly without too much typing. And before that, we need to prepare a dataset with ROW variables:

```

data D20(keep=USUBJID ROW:);
  set D10;
  if GENDER=1 then ROW001=1;
  if GENDER=2 then ROW002=1;
run;

```

After running this macro, we can get dataset F with all the countings we requested.

In certain situation, we need to count number of unique patients. To do that, we need remove duplicates first. We can add one procedure in the macro we mentioned above:

```

proc sort data=D20(where=(ROW&var=1)) out=D30 nodupkey;
  by USUBJID;
run;

```

Because we are only interested in one condition, we can keep the dataset as simple as possible, removing unnecessary values and duplicates.

## MEANS FOR CONTINUOUS VARIABLES

If variables are continuous, we usually need to compute means, standard derivation, medians, min and max, confidence limits, and ranges, etc. For example:

	Drug	Placebo	Total
Age			
N	xx (xx.x%)	xx (xx.x%)	xx (xx.x%)
Mean (SD)	xx.x (xx.xx%)	xx.x (xx.xx%)	xx.x (xx.xx%)
Median	xx.x	xx.x	xx.x
Q1, Q3	xx.x, xx.x	xx.x, xx.x	xx.x, xx.x
95% CI	xx.x, xx.x	xx.x, xx.x	xx.x, xx.x
Min, Max	xx, xx	xx, xx	xx, xx

We will use proc means to compute these values. Here we introduce a simple macro:

```

%macro _means(outds=,var=);
  proc means data=D20 noprint;
    var &var;
    output out=&outds(drop=_TYPE_ _FREQ_)
      N=N MEAN=MEAN STD=SD MEDIAN=MEDIAN Q1=Q1 Q3=Q3
      MIN=MIN MAX=MAX LCLM=LCLM UCLM=UCLM;
  run;

  data &outds;

```

```
        set &outds;  
        Row=input(substr("&outds",2),best.);  
    run;  
%mend;
```

Again, it is essential to use it efficiently. For example:

```
%_means(outds=M01,var=AGE);
```

Please note here we also have a variable ROW, which works with the ROW in counts to contribute to the final table.

## CONCLUSION

Here we introduce two simple macros to help producing descriptive statistic summary tables. They are highly generalized so that they can be used in almost all the counts and means tables. Therefore, we do not need to take too much time think about how to compute each variable and each condition. As a result, with this method, we can even create a table, no matter how big it might be, in half an hour. They will not only save a lot of time but also improve the quality.