

## An Easy-to-use SAS® Macro for a Descriptive Statistics Table

Yuanchao Zheng, Stanford University; Jin Long, Stanford University;

Maria E. Montez-Rath, Stanford University

### ABSTRACT

Are you tired of copying output from the Proc Freq or Proc Means procedures and pasting it into your tables? Do you need to produce summary tables repeatedly? Are you spending a lot of your time generating the same summary tables for different subpopulations? This paper introduces an easy-to-use macro to generate a descriptive statistics table. The table reports counts and percentages for categorical variables as well as means, standard deviations, medians, and quantiles for continuous variables. For variables with missing values, the table also includes the count and percentage missing. Customization options allow for the analysis of stratified data, the specification of variables' output order, and user-defined formats. Additionally, this macro incorporates the SAS Output Delivery System (ODS) to automatically output a Rich Text Format (RTF) file, which can be further edited by a word processor for the purpose of publication.

### INTRODUCTION

Summary tables are used frequently to describe data, present the characteristics of variables of interest, or show the differences among study populations. Even though Proc Means and Proc Freq procedures are extremely useful SAS procedures that allow the user quick access to descriptive summary statistics, organizing multiple results from their outputs into one summary table can take a lot of effort. Repetitive copying and pasting are usually required. Furthermore, any changes made to the population can easily cause a doubling of work. Hence, it is important to develop an easy-to-use SAS macro that automatically generates a customized table with descriptive statistics. In the sections that follow, we provide full instructions and give examples that show how our macro can be used for this purpose.

### INSTRUCTIONS

The macros in the Appendix need to be saved to your computer. Afterwards, they can be included in your SAS program using the %include statement. The following table lists each of the options and its description.

Options	Description
<b>Required</b>	
%let yourdata=;	Name of SAS data set containing variables to be summarized.
%let output_data=;	Name of SAS data set containing summarized statistics. This is also the name of output RTF file.
%let decimal_max=;	Specify how many decimal points you need: 0, 1, 2, 3. This option does not apply to count data.
%let varlist_cat=;	List of categorical variables. Leave empty if none.
%let varlist_cont=;	List of continuous variables. Leave empty if none.
<b>Optional</b>	
%let formatsfolder=;	Location of SAS formats. Leave empty if none. When specified, include quotations, e.g., "C:\formats".
%let yourfolder=;	Location where your data set is saved. Leave empty for the SAS work library. When specified, include quotations, e.g., "C:\data".
%let output_order=;	List of all UNIQUE variables from varlist_cat and varlist_cont in the order to

	be shown in the output table. Leave empty for default order, i.e., order entered in <i>varlist_cont</i> and <i>varlist_cat</i> .
<code>%let group_by=;</code>	Specify whether you want to output results by categories, e.g., gender. Formatted values will be used as priority, then raw values. Leave empty to obtain statistics for the whole population. If a group-by variable is specified, a category for unformatted missing data can be created by user's option. See the <i>group_by_missing</i> option below.
<code>%let group_by_missing=;</code>	Specify whether or not output statistics for those observations with unformatted missingness in the group-by variable: 0, 1. Required if the <i>group_by</i> option is used. Value 1 creates a category for missing group-by variable. Change to 0 if not interested in reporting summary statistics for those missingness.

## DATA EXAMPLE

To illustrate, we can create a SAS data set called *TESTDATA*, and apply formats as appropriate.

```

data rawdata;
  input ID sex $ age race BMI;
  datalines;
1      F      19      1      18.5
2      .      38      1      19.5
3      F      72      1      25.1
4      F      21      1      22.0
5      .      20      2      33.2
6      F      24      2      17.5
7      F      28      2      28.2
8      F      33      2      29.5
9      F      .       2      25.3
10     F      65      2      29.2
11     .      77      1      20.5
12     F      .       1      24.6
13     F      21      2      19.5
14     F      27      1      18.5
15     F      39      1      18.4
16     M      55      2      19.0
17     M      71      1      21.7
18     M      .       1      34.4
19     .      35      1      26.3
20     M      28      1      22.9
21     M      31      1      27.3
22     M      29      2      23.7
23     M      25      2      32.1
24     .      26      1      25.3
25     M      38      1      16.8
;
run;

```

```

proc format;
  value $gender
    "F"="Female"
    "M"="Male"
  ;
  value race_cat
    1="White"
    2="Black"
  ;
  value age_cat
    low-40="<=40"
    40<-high=">40"
    .="Missing age"
  ;
run;

data testdata;
  set rawdata;
  format sex gender. race race_cat. age age_cat.;
run;

```

### EXAMPLE 1: CHECK DESCRIPTIVE STATISTICS FOR THE WHOLE POPULATION

Save the macros in the Appendix in your computer as in "C:\your folder\summary\_stat\_macro.sas". Run the code below to get summary statistics for age, sex and race for the whole population.

In this example, we want statistics for the whole population, so the *group\_by* and *group\_by\_missing* options are not specified. Because we used a format to categorize age, we can obtain statistics for both age categorical and age continuous simply by entering age in both lists: *varlist\_cat* and *varlist\_cont*.

```

%include "C:\your folder\summary_stat_macro.sas";

%let yourdata=testdata;           /*name of your SAS data set*/
%let output_data=test_summary1;   /*name of output SAS data set*/

%let formatsfolder=;             /*location of your SAS formats*/
%let yourfolder=;                /*location of your SAS data set*/

%let varlist_cat = age race sex;  /*list of categorical variables*/
%let varlist_cont = age;         /*list of continuous variables*/
%let output_order = age race sex; /*output order of all UNIQUE variables*/

%let decimal_max=1;              /*desired number of decimal points*/

%let group_by=;                  /*name of stratification variable*/
%let group_by_missing=;         /*whether to remove observations missing
                                the stratification variable.*/

%Table_summary;                  /*call the macros*/

```

SAS ODS automatically outputs an RTF file named as "test\_summary1.rtf", with the following descriptive statistics table.

Variable	Description	Type of statistic	All (n=25)
AGE	Continuous	mean and std	37.4 (18.3)
AGE	Continuous	median and IQR	30.0 (25.0, 39.0)
AGE	Continuous	min and max	(19.0, 77.0)
AGE	Continuous: missing	count and percent	3 (12.0%)
AGE	Missing age	count and percent	3 (12.0%)
AGE	<=40	count and percent	17 (68.0%)
AGE	>40	count and percent	5 (20.0%)
RACE	White	count and percent	15 (60.0%)
RACE	Black	count and percent	10 (40.0%)
SEX	Missing	count and percent	5 (20.0%)
SEX	Female	count and percent	12 (48.0%)
SEX	Male	count and percent	8 (32.0%)

Interpretation of table:

- The column “Variable” shows the variable name the statistics in each row refer to. The column “Description” indicates whether the variable is being treated as a continuous variable or, if categorical, the category name. The column “Type of statistic” indicates the type of statistic being reported.
- The last column is labelled as “All (n=25)”, indicating that the total number of observations used to create the descriptive statistics is 25.
- Age is treated as both continuous and categorical. For continuous variables, the macro computes statistics using those non-missing observations, and reports count and percentage of missing observations. For categorical variables, the macro computes statistics including missing observations.

## EXAMPLE 2: CHECK DESCRIPTIVE STATISTICS FOR A STRATIFIED POPULATION

To obtain descriptive statistics stratified by sex, specify sex in the *group\_by* option. By specifying the *group\_by\_missing* option to be 1, observations with missing sex will be output as a separate category.

SAS ODS automatically outputs a new RTF file named as “test\_summary2.rtf”.

```

%let yourdata=testdata;           /*name of your SAS data set*/
%let output_data=test_summary2;   /*name of output SAS data set*/

%let formatsfolder=;             /*location of your SAS formats*/
%let yourfolder=;               /*location of your SAS data set*/

%let decimal_max=1;              /*desired number of decimal points*/

%let varlist_cat = race;         /*list of categorical variables*/
%let varlist_cont = age;        /*list of continuous variables*/
%let output_order = age race;    /*output order of all UNIQUE variables*/

%let group_by=sex;               /*name of stratification variable*/
%let group_by_missing=1;        /*keep observations missing the
                                stratification variable.*/

%Table_summary;                 /*call the macros*/

```

Variable	Description	Type of statistic	ALL (n=25)	FEMALE (n=12)	MALE (n=8)	UNSPECIFIED MISSING (n=5)
AGE	Continuous	mean and std	37.4 (18.3)	34.9 (18.8)	39.6 (17.1)	39.2 (22.3)
AGE	Continuous	median and IQR	30.0 (25.0, 39.0)	27.5 (21.0, 39.0)	31.0 (28.0, 55.0)	35.0 (26.0, 38.0)
AGE	Continuous	min and max	(19.0, 77.0)	(19.0, 72.0)	(25.0, 71.0)	(20.0, 77.0)
AGE	Continuous: missing	count and percent	3 (12.0%)	2 (16.7%)	1 (12.5%)	
RACE	Black	count and percent	10 (40.0%)	6 (50.0%)	3 (37.5%)	1 (20.0%)
RACE	White	count and percent	15 (60.0%)	6 (50.0%)	5 (62.5%)	4 (80.0%)

To exclude observations with missing sex in the table, you can change the *group\_by\_missing* option to be 0. In this case, a footnote is provided together with the summary table in the RTF output, indicating that number of observations dropped due to unformatted missingness of the group-by variable. In this example, 5 observations will be deleted due to missing in the variable sex.

```

%let yourdata=testdata;           /*name of your SAS data set*/
%let output_data=test_summary3;   /*name of output SAS data set*/

%let formatsfolder=;             /*location of your SAS formats*/
%let yourfolder=;                /*location of your SAS data set*/

%let decimal_max=1;              /*desired number of decimal points*/

%let varlist_cat = race;         /*list of categorical variables*/
%let varlist_cont = age;         /*list of continuous variables*/
%let output_order = age race;    /*output order of all UNIQUE variables*/

%let group_by=sex;               /*name of stratification variable*/
%let group_by_missing=0;         /*remove observations missing the
                                stratification variable.*/

%Table_summary;                 /*call the macros*/

```

Variable	Description	Type of statistic	ALL (n=20)	FEMALE (n=12)	MALE (n=8)
AGE	Continuous	mean and std	36.8 (17.7)	34.9 (18.8)	39.6 (17.1)
AGE	Continuous	median and IQR	29.0 (25.0, 39.0)	27.5 (21.0, 39.0)	31.0 (28.0, 55.0)
AGE	Continuous	min and max	(19.0, 72.0)	(19.0, 72.0)	(25.0, 71.0)
AGE	Continuous: missing	count and percent	3 (15.0%)	2 (16.7%)	1 (12.5%)
RACE	Black	count and percent	9 (45.0%)	6 (50.0%)	3 (37.5%)
RACE	White	count and percent	11 (55.0%)	6 (50.0%)	5 (62.5%)

Another way to handle the missing data in the group-by variable, is by creating a format that includes an option for the missing data and apply the format to the group-by variable. For example:

```

proc format;
  value $gender_new
    "F"="Female"
    "M"="Male"
    " "="Missing Gender"
  ;
run;

data testdata2;
  set rawdata;
  format sex $gender_new. race race_cat. age age_cat.;
run;

%let yourdata=testdata2;           /*name of your SAS data set*/
%let output_data=test_summary4;    /*name of output SAS data set*/

%let formatsfolder=;              /*location of your SAS formats*/
%let yourfolder=;                 /*location of your SAS data set*/

%let decimal_max=1;               /*desired number of decimal points*/

%let varlist_cat = race;           /*list of categorical variables*/
%let varlist_cont = age;           /*list of continuous variables*/
%let output_order = age race;     /*output order of all UNIQUE variables*/

%let group_by=sex;                 /*name of stratification variable*/
%let group_by_missing=0;           /*remove observations missing the
                                     stratification variable.*/

%Table_summary;                   /*call the macros*/

```

Variable	Description	Type of statistic	ALL (n=25)	FEMALE (n=12)	MALE (n=8)	MISSING GENDER (n=5)
AGE	Continuous	mean and std	37.4 (18.3)	34.9 (18.8)	39.6 (17.1)	39.2 (22.3)
AGE	Continuous	median and IQR	30.0 (25.0, 39.0)	27.5 (21.0, 39.0)	31.0 (28.0, 55.0)	35.0 (26.0, 38.0)
AGE	Continuous	min and max	(19.0, 77.0)	(19.0, 72.0)	(25.0, 71.0)	(20.0, 77.0)
AGE	Continuous: missing	count and percent	3 (12.0%)	2 (16.7%)	1 (12.5%)	
RACE	Black	count and percent	10 (40.0%)	6 (50.0%)	3 (37.5%)	1 (20.0%)
RACE	White	count and percent	15 (60.0%)	6 (50.0%)	5 (62.5%)	4 (80.0%)

## TIPS

- If your data is saved externally under "C:\data folder" and the external formats are stored under "C:\formats folder", you can specify the following options:

```
%let yourfolder = "C:\data folder";
```

```
%let formatsfolder = "C:\formats folder".
```

- Continuous variables can be included in both *varlist\_cat* and *varlist\_cont* at the same time, but should only appear once in the *output\_order* (e.g. the AGE variable in EXAMPLE 1).
- Addition to the output RTF file, an output SAS data containing summarized statistics will be generated, which has an additional column for variable labels.

## CONCLUSION

We provide a macro that is designed for any SAS user interested in producing summary tables with descriptive statistics. The macro has a flexible design and can be easily included in your SAS programs. Its ODS RTF output with editable tables can save time and help reduce errors when compared with the usual way of copying and pasting SAS output. Moreover, with customizable SAS Formats, you can easily create well organized tables that describe your data.

## ACKNOWLEDGMENTS

We greatly appreciate the support of our colleagues Michelle M. O'Shaughnessy, Sai Liu, Tara I. Chang, Margaret R. Stedman and Jinnie Rhee at the Division of Nephrology, Stanford School of Medicine, who tested the macro and provided valuable feedback.

## RECOMMENDED READING

- *SAS® Certification Prep Guide: Advanced Programming for SAS®9*

## CONTACT INFORMATION

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

Yuanchao Zheng  
Division of Nephrology, Department of Medicine  
Stanford University School of Medicine  
1070 Arastradero Rd., Suite 100  
Palo Alto, CA 94304  
Email: [yuanchao@stanford.edu](mailto:yuanchao@stanford.edu)

## APPENDIX

```

/*****
*****
*** Author: Yuanchao Zheng, M.S., Stanford University
*** Purpose: Create descriptive tables
*** OS: Windows 10 Pro 64-bit
*** Software: SAS 9.4

*** Note:
    There are two SAS macros: Getvars and Table_summary. Getvars computes
    summary statistics for both categorical and continuous variables.
    Table_summary calls Getvars and outputs a summary table.

*** Copyright (C) <2017> <Yuanchao Zheng>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.

*****/
/*****
*****/
/*****
*** Marco 1: Getvars;
*****/
%macro Getvars(dat=&yourdata.,datout=&output_data.);

proc datasets; delete &datout.;run; quit;

*Get summary statistics for categorical variables;
*****;
%if &varlist_cat.^= %then %do;

*count number of categorical variables;
%let nvarlist_cat=%sysfunc(countw(&varlist_cat.));

*for each categorical variable, compute counts and percents;
%do i=1 %to &nvarlist_cat.;
%let var&i.=%scan(&varlist_cat.,&i.);

proc freq data=&dat. noprint; tables &&var&i../missing out=dat_cat_&i.; run;

data dat_cat_&i._2(drop=&&var&i.. COUNT PERCENT);
set dat_cat_&i.;

```



```

length xx_variable xx_var_label xx_description xx_type xx_var
xx_variable_original $ 100;

output_subborder=5;
xx_variable="%trim(%upcase(&&var&i..))";
xx_var_label=upcase(VLABEL(&&var&i..));
xx_description=strip(vvalue(&&var&i..));
xx_type="count and percent";
xx_var=strip(put(COUNT,16.))||" ("||strip(put(PERCENT,
16.&decimal_max.))||"%")";
xx_variable_original="%trim(%upcase(&&var&i..))";

label xx_variable="Variable" xx_var_label="Variable label"
      xx_description="Description" xx_type="Type of statistic"
      xx_var="Statistic";
format percent 8.1;
run;

proc append base=&datout. data=dat_cat_&i._2; run;

%end;

proc datasets library=work; delete dat_cat:; run;

%end;

*Get summary statistics for continuous variables;
*****;

*part 1: mean and std values;

%if &varlist_cont.^= %then %do;

%let nvarlist_cont=%sysfunc(countw(&varlist_cont.));

%do j=1 %to &nvarlist_cont.;
%let var&j.=%scan(&varlist_cont.,&j.);

proc means data=&dat. noprint; var &&var&j..;
output out=dat_cont1_&j.(drop=_TYPE_ _FREQ_) mean=mean std=std;
run;

data dat_cont1_&j._2(drop=mean std);
set dat_cont1_&j.;
length xx_variable xx_var_label xx_description xx_type xx_var
xx_variable_original $ 100;

output_subborder=1;
xx_variable="%trim(%upcase(&&var&j..))";
xx_var_label=strip(upcase(VLABEL(mean)));
xx_description="Continuous";
xx_type="mean and std";
xx_var=strip(put(mean,16.&decimal_max.))||" ("||strip(put(std,
16.&decimal_max.))||")";
xx_variable_original="%trim(%upcase(&&var&j..))";
if xx_var_label="MEAN" then xx_var_label=xx_variable;

```

```

label xx_variable="Variable" xx_var_label="Variable label"
      xx_description="Description" xx_type="Type of statistic"
      xx_var="Statistic";
run;

proc append base=&datout. data=dat_cont1_&j._2; run;

*part 2: median and IQR, min max;

proc means data=&dat. noprint; var &&var&j..;
  output out=dat_cont2_&j.(drop=_TYPE_ _FREQ_)
         p50=p50 p25=p25 p75=p75 min=min max=max;
run;

data dat_cont2_&j._2(drop=p50 p25 p75);
  set dat_cont2_&j.(drop=min max);
  length xx_variable xx_var_label xx_description xx_type xx_var
xx_variable_original $ 100;

  output_suborder=2;
  xx_variable="%trim(%upcase(&&var&j..))";
  xx_var_label=strip(upcase(VLABEL(p50)));
  xx_description="Continuous";
  xx_type="median and IQR";
  xx_var=strip(put(p50,16.&decimal_max.)||" ("||strip(put(p25,
16.&decimal_max.)||", "||strip(put(p75, 16.&decimal_max.)||")");
  xx_variable_original="%trim(%upcase(&&var&j..))";
  if xx_var_label="P50" then xx_var_label=xx_variable;

  label xx_variable="Variable" xx_var_label="Variable label"
        xx_description="Description" xx_type="Type of statistic"
        xx_var="Statistic";
run;

data dat_cont2_&j._3(drop=min max);
  set dat_cont2_&j.(drop=p50 p25 p75);
  length xx_variable xx_var_label xx_description xx_type xx_var
xx_variable_original $ 100;

  output_suborder=3;
  xx_variable="%trim(%upcase(&&var&j..))";
  xx_var_label=strip(upcase(VLABEL(min)));
  xx_description="Continuous";
  xx_type="min and max";
  xx_var="("||strip(put(min, 16.&decimal_max.)||", "||strip(put(max,
16.&decimal_max.)||")");
  xx_variable_original="%trim(%upcase(&&var&j..))";
  if xx_var_label="MIN" then xx_var_label=xx_variable;

  label xx_variable="Variable" xx_var_label="Variable label"
        xx_description="Description" xx_type="Type of statistic"
        xx_var="Statistic";
run;

proc append base=&datout. data=dat_cont2_&j._2; run;
proc append base=&datout. data=dat_cont2_&j._3; run;

```

```

*part 3: check missingness;
proc format;
  value $mchar
    " " = "Missing"
    other = "Not Missing"
  ;
  value mnum
    . = "Missing"
    other = "Not Missing"
  ;
run;

data dat_cont_check_&j.;
  set &dat.(keep=&&var&j.);
  format _CHAR_ $mchar. _NUMERIC_ mnum.;
run;

proc freq data=dat_cont_check_&j. noprint;
  tables &&var&j../missing out=dat_cont_check_&j._2;
run;

data dat_cont_check_&j._3(drop=&&var&j.. COUNT PERCENT);
  set dat_cont_check_&j._2;
  length xx_variable xx_var_label xx_description xx_type xx_var
  xx_variable_original $ 100;

  if vvalue(&&var&j..) = "Missing";

  output_suborder=4;
  xx_variable = "%trim(%upcase(&&var&j..))";
  xx_var_label = upcase(VLABEL(&&var&j..));
  xx_description = "Continuous: missing";
  xx_type = "count and percent";
  xx_var = strip(put(COUNT,16.)) || " (" || strip(put(PERCENT,
16.&decimal_max.)) || "%" || ")";
  xx_variable_original = "%trim(%upcase(&&var&j..))";

  label xx_variable = "Variable" xx_var_label = "Variable label"
  xx_description = "Description" xx_type = "Type of statistic"
  xx_var = "Statistic";
  format percent 8.1;
run;

proc append base=&datout. data=dat_cont_check_&j._3; run;
%end;

proc datasets library=work; delete dat_cont:; run;
%end;
%mend;

/*****
*** Marco 2: Table_summary;
*****/

%macro Table_summary(byvar=&group_by.);

```

```

/*****SAS settings*****/

options missing="" PAGENO=1 label nofmterr formdlm=" ";

*Set SAS formats if there is any external one;
%if &formatsfolder.^= %then %do; libname library &formatsfolder.;%end;

*Set where data to be summarized is stored. Default is SAS work library.;
%if &yourfolder.^= %then %do;
    libname datain &yourfolder.;
    data &yourdata.; set datain.&yourdata.; run;
%end;
%else %do; data &yourdata.; set work.&yourdata.; run; %end;

*If none of variable is specified as categorical or continuous.;
%if &varlist_cat.= and &varlist_cont.= %then %do;
    %put ERROR: no variable is specified.; %return;
%end;

*If no variable output order is specified, then use the default order.;
%if &output_order.= %then %do;
    %let output_order=%sysfunc(catx(%str( ),&varlist_cont., &varlist_cat.));
%end;

/*****If group_by option is NOT used*****/

%if &byvar.= %then %do;

*Count the total number of rows;
proc sql; select count(*) into: n_population from &yourdata.; quit;
%let n_population = &n_population.;

*Call the macro %getvars to get all statistics;
%getvars(dat=&yourdata., datout=&output_data.)

data &output_data.;
    set &output_data.(rename=(xx_var=xx_var_all));
    label xx_var_all="All (n=%trim(&n_population.))";
run;

*Create a dataset that contains variable output orders.;
proc datasets; delete table_orders; run;
%let noutput_order=%sysfunc(countw(&output_order.));

data table_orders;
    length xx_variable_original $ 100 output_order 8;
    xx_variable_original=" "; output_order=0;
    output;
run;

%do tableorder=1 %to &noutput_order.;
%let tableorder_var&tableorder.=%scan(&output_order.,&tableorder.);
data table_orders;
    set table_orders end=eof;
    length xx_variable_original $ 100 output_order 8; output;
    if eof then do;
        xx_variable_original="%trim(%upcase(&&tableorder_var&tableorder.))";
    end;
%end;

```

```

        output_order=&tableorder.;
    output;
end;
run;
%end;

data table_orders; set table_orders(where=(output_order^=0)); run;

*Add variable output orders to dataset with all statistics;
*Sort data by specified orders;
proc sort data=&output_data.; by xx_variable_original output_suborder;
proc sort data=table_orders nodupkey; by xx_variable_original; run;
data &output_data.;
    merge &output_data.(in=a) table_orders;
    by xx_variable_original;
    if a;
run;
proc datasets; delete table_orders; run;

proc sort data=&output_data.; by output_order output_suborder; run;
data &output_data.;
    set &output_data.(drop=output_order output_suborder xx_variable_original);
    if indexc(xx_var_all, '0123456789')=0 then xx_var_all="";
    if missing(xx_description) then xx_description="Missing";
    if xx_variable=xx_var_label then xx_var_label=" ";
run;

*Output the final summary table;
ods listing close;
options missing="" orientation=landscape number date;
ods rtf bodytitle style=statistical file="&output_data..rtf";
footnote;
title "Descriptive Table";

proc print data=&output_data. label;
    var xx_variable xx_description xx_type xx_var_all;
run;

ods rtf close;
title; footnote;

%end;

/*****If group_by option is used*****/

%else %do;

*Create a character variable to represent original group-by variable;
*Formatted values will be used if the original variable contains formats.;
data yourdata2;
    set &yourdata.;
    length &byvar._temp $ 50;

*Create a character variable to represent original group-by variable;
&byvar._temp=strip(vvalue(&byvar.));

*Whether or not delete unspecified missing values in group-by variable;

```

```

%if &group_by_missing.=0 %then %do;
  if missing(&byvar._temp) then delete;
%end;
%else %if &group_by_missing.=1 %then %do;
  if missing(&byvar._temp) then &byvar._temp="Unspecified Missing";
%end;
run;

%let byvar_old=%trim(&byvar.);
%let byvar=&byvar._temp;

*compute the number of different subpopulations in group-by variable;
proc freq data=yourdata2(keep=&byvar.) noprint;
  tables &byvar./out=temp_byvar;
run;
proc sql; select count(*) into: n_byvar from temp_byvar; quit;
%let n_byvar = &n_byvar.;

*for each subpopulation, compute the number of observations;
data _NULL_;
  length byvar_formatted obs $ 100;
  set temp_byvar;
  byvar_formatted=vvalue(&byvar.);
  obs="value"||strip(_N_);
  obs2="counts"||strip(_N_);
  call symput(obs,trim(byvar_formatted)); call symput(obs2,trim(COUNT));
run;

*for each subpopulation, compute all the statistics;
proc datasets; delete table_suborders; run;

%do m=1 %to &n_byvar.;

data temp_data_sub&m.;
  set yourdata2;
  if strip(uppercase(vvalue(&byvar.)))=strip(uppercase("&&value&m.."));
run;

%getvars(dat=temp_data_sub&m., datout=temp_Cohort_demo_sub&m.)

data temp_Cohort_demo_sub&m.;
  set temp_Cohort_demo_sub&m.(rename=(xx_var=xx_var&m.));
  if indexc(xx_var&m., '0123456789')=0 then xx_var&m.="";
  label xx_var&m.="%quppercase(&&value&m..) (n=%trim(&&counts&m..))";
run;

proc sort data=temp_Cohort_demo_sub&m.;
  by xx_variable_original xx_variable xx_var_label xx_description xx_type;
run;

proc append base=table_suborders
  data=temp_Cohort_demo_sub&m.(keep=xx_variable_original
xx_variable xx_var_label xx_description xx_type output_suborder);
run;
%end;

*compute the number of total observations, and statistics for all;

```

```

proc sql;select count(&byvar.) into: n_population from yourdata2; quit;
%let n_population = &n_population.;

%getvars(dat=yourdata2, datout=temp_cohort_demo_all)

data temp_cohort_demo_all(drop=output_suborder rename=(xx_var=xx_var_all));
  set temp_cohort_demo_all;
  if indexc(xx_var, '0123456789')=0 then xx_var="";
  label xx_var="ALL (n=%trim(&n_population.))";
run;

*Create a dataset that contains variable output orders.;
proc datasets; delete table_orders; run;
%let noutput_order=%sysfunc(countw(&output_order.));
data table_orders;
  length xx_variable_original $ 100 output_order 8;
  xx_variable_original=" "; output_order=0; output;
run;

%do tableorder=1 %to &noutput_order.;
%let tableorder_var&tableorder.=%scan(&output_order.,&tableorder.);
data table_orders;
  set table_orders end=eof;
  length xx_variable_original $ 100 output_order 8;
  output;
  if eof then do;
    xx_variable_original="%trim(%upcase(&tableorder_var&tableorder.))";
    output_order=&tableorder.;output;
  end;
run;
%end;
data table_orders; set table_orders(where=(output_order^=0)); run;

proc sort data=table_orders nodupkey; by xx_variable_original; run;
proc sort data=table_suborders nodup;
  by xx_variable_original xx_variable xx_var_label xx_description xx_type;
run;

data orders;
  merge table_orders(in=a) table_suborders;
  by xx_variable_original;
  if a;
run;

*Add variable output orders to dataset with all statistics;
proc sort data=orders nodup;
  by xx_variable_original xx_variable xx_var_label xx_description xx_type;
run;

proc sort data=temp_cohort_demo_all;
  by xx_variable_original xx_variable xx_var_label xx_description xx_type;
run;

data &output_data.;

%if n_byvar.>1 %then %do;
merge orders(in=a) temp_cohort_demo_all

```

```

        temp_Cohort_demo_sub1 - temp_Cohort_demo_sub&n_byvar.;
    %end;
    %else %if n_byvar.=1 %then %do;
    merge orders(in=a) temp_cohort_demo_all temp_Cohort_demo_sub1;
    %end;

    by xx_variable_original xx_variable xx_var_label xx_description xx_type;
    if a;
run;

proc sort data=&output_data.; by output_order output_suborder; run;
data &output_data.(drop=xx_variable_original output_order output_suborder);
    set &output_data.;
    if missing(xx_description) then xx_description="Missing";
    if xx_variable=xx_var_label then xx_var_label=" ";
run;

*Check whether there is any missing value in the group-by variable.;
proc sql;
    select count(*) into: nmiss1 from &yourdata.;
    select count(*) into: nmiss2 from yourdata2;
quit;
%let nmiss=%eval(&nmiss1.-&nmiss2.);
%let nmiss=%trim(&nmiss.);

*Delete unnecessary datasets;
proc datasets library=work;
    delete orders table_orders table_suborders yourdata2 temp;;
run;

*Output the final summary table;
ods listing close;
options missing="" orientation=landscape number date;
ods rtf bodytitle style=statistical file="&output_data..rtf";

title "Descriptive Table by Variable &byvar_old.";
%if &group_by_missing.=0 %then %do;
    footnote "NOTE: there are &nmiss. observations deleted due to missing
values in the variable &byvar_old.";
%end;

proc print data=&output_data. label;
%if n_byvar.>1 %then %do;
    var xx_variable xx_description xx_type xx_var_all xx_var1-xx_var&n_byvar.;
    %end;
%else %if n_byvar.=1 %then %do;
    var xx_variable xx_description xx_type xx_var_all xx_var1;
    %end;
run;

ods rtf close;
title;footnote;
%end;
%mend;

```