Paper 97-26

Side By Side: Comparing Two Data Sets Using SAS® Macros

Aileen D. Bennett, U.S. Census Bureau

ABSTRACT

This paper describes a generalized program to compare two SAS data sets with similar names which contain the same variables. The user specifies data set names and libraries. The program uses SAS macros to generate frequencies or means of numeric variables common to the data sets and print them side by side. The paper is intended for intermediate/advanced users of SAS who are familiar with macros.

INTRODUCTION

Each month the U.S. Census Bureau conducts the Current Population Survey for the Bureau of Labor Statistics to gather labor force data. The survey frequently includes supplemental questions on various topics. The questions are asked of roughly the same number of people each time, so one would expect to see about the same numbers and distributions of responses. Analysts wanted to compare and contrast the data from two instances of a supplement by looking at frequencies from two data sets side by side. The resulting program uses SAS macros to produce side-by-side frequencies for any two supplement months.

MACRO VARIABLES AND FORMAT

Set the macro variables for the months and years desired, the data set names, and libraries. Set format values for printing the mean and universe size labels:

```
%let mmm1 = jan;
%let yy1 = 96;
%let mmm2 = may;
%let yy2 = 96;
%let dsname = inp;
libname in1 "/supp6/&mmm1&yy1";
libname in2 "/supp6/&mmm2&yy2";
title "&mmm1&yy1 - &mmm2&yy2 Comparison";

proc format;
 value notefmt -88 = 'Mean'
               -99 = 'Universe Size';
run;
```

DATA SET CONTENTS

Create data sets containing the contents of each data set. They should contain the same variables, but might not completely, as supplements sometimes change slightly from year to year.

```
proc contents data=in1.&mmm1&yy1&dsname
 out=cont1 noprint;
 run;
proc sort data=cont1(keep=name type);
 by name;
 run;

proc contents data=in2.&mmm2&yy2&dsname
 out=cont2 noprint;
 run;
proc sort data=cont2(keep=name type);
 by name;
 run;
```

GENERATED MACRO VARIABLES

Create a macro variable for each variable to be compared. Only numeric variables whose name has "U" for the second character will be included. For each variable, create two flags indicating whether or not it is in each data set. Create a macro variable containing a count of the variables:

```
data _null_;
 retain count 0;
 merge cont1(in=in1 rename=(type=type1))
       cont2(in=in2 rename=(type=type2))
       end=last;
 by name;
 if type1=1 & type2=1 & substr(name,2,1)='U'
  then do;
  count + 1;
  call
   symput('varnm'||left(put(count,3.)),name);
  if in1 then call
   symput('in1'||left(put(count,3.)),'y');
  else call
   symput('in1'||left(put(count,3.)),'n');
  if in2 then call
   symput('in2'||left(put(count,3.)),'y');
  else call
   symput('in2'||left(put(count,3.)),'n');
 end;
 if last then call
   symput('count',put(count,3.));
 run;
```

MAIN MACRO

The RUNCOMP macro executes once for each compared variable. It checks to see if a variable is in one or both data sets. If it is in both data sets, the COMPARE macro is called with the variable name parameter. If it is in only one data set, the IN1 macro is called with the variable name and data set number parameters:

```
%macro runcomp;
 %do i = 1 %to &count;
  %if &&in1&i=y & &&in2&i=y %then %do;
    %compare(&&varnm&i)
  %end;
  %else %if &&in1&i=y & &&in2&i=n %then %do;
   %in1(&&varnm&i,1)
  %end;
  %else %if &&in1&i=n & &&in2&i=y %then %do;
   %in1(&&varnm&i,2)
  %end;
  %else %put '#Error ' &&varnm&i &&in1&i
    &&in2&i;
 %end;
%mend runcomp;
```

COMPARE MACRO

Calculate the universe size of the variable in each data set. This is the count of the number of responses which are not blank (represented by -1):

```
%macro compare(var);
%do j = 1 %to 2;
 proc summary nway data=
  in&j..&&mmm&j.&&yy&j.&&dsname
  (where=(&var^=-1));
  var &var;
  output out=univ&j n=total&j;
  run;
%end;

data univ;
 drop _type_ _freq_;
```

```
    merge univ1 univ2;
    run;
```

Create a frequency of the variable from the first data set and put it into a data set:

```
  proc freq data=in1.&mmm1&yy1&dsname;
    tables &var/noprint out=freq1;
       run;
```

Count the number of frequency values and put it into a macro variable:

```
  proc sql noprint;
   select nobs into :freqcnt from
   dictionary.tables
   where libname='WORK' & memname='FREQ1';
   quit;
```

If there are 20 or fewer values, the variable is considered discrete: for example, a question with a yes/no response. For discrete variables, create a frequency of the variable from the second data set and merge it with the frequency from the first data set:

```
  %if &freqcnt <= 20 %then %do;
   proc freq data=in2.&mmm2&yy2&dsname;
    tables &var/noprint out=freq2;
    run;

   data freq;
    merge freq1(rename=count=count1
                       percent=percent1)
          freq2(rename=count=count2
                       percent=percent2));
    by &var;
    run;
```

Add the universe sizes:

```
   data all;
    drop total1 total2;
    set freq univ(in=a);
    if a then do;
     &var = -99;
     count1 = total1;
     count2 = total2;
    end;
    label count1 = "&mmm1&yy1"
          count2 = "&mmm2&yy2";
    run;
```

Print the frequencies side by side:

```
  proc print noobs label;
   var &var count1 count2;
   format &var notefmt.;
   run;
 %end;
```

jan96 - may96 Comparison

PUS32	jan96	may96
-4	73	82
-3	717	658
-2	266	243
-1	13651	11497
1	34766	34447
2	42932	46038
Universe Size	78754	81468

If the frequency has more than 20 values, the variable is continuous. An example is the response to the question "At what age did you...?". For continuous variables, the program compares the means. Calculate the mean of the variable from each data set and merge the means into one data set:

```
  %else %do;
   %do j = 1 %to 2;
    proc summary nway
         data=in&j..&&mmm&j.&&yy&j.&&dsname
         (where=(&var>0));
     var &var;
     output out=summout&j mean=mean&j;
     run;
   %end;

   data summout;
    merge summout1 summout2;
    run;
```

Add the universe sizes:

```
   data all;
    drop total1 total2;
    set summout(in=a) univ(in=b);
    if a then &var = -88;
    else if b then do;
     &var = -99;
     mean1 = total1;
     mean2 = total2;
    end;
    label mean1 = "&mmm1&yy1"
          mean2 = "&mmm2&yy2";
    run;
```

Print the means side by side:

```
  proc print noobs label;
   var &var mean1 mean2;
   format &var notefmt.;
   run;
 %end;
%mend compare;
```

jan96 - may96 Comparison

PUS33	Jan96	May96
Mean	17.86	17.83
Universe Size	34841.00	34503.00

IN1 MACRO

The IN1 macro is called if the variable is in only one of the data sets, so comparison is not possible. It prints a PROC FREQ or PROC MEANS of the variable, depending on whether the variable is discrete or continuous as defined above:

```
   %macro in1(var,i);
   proc freq data=in&i..&&mmm&i.&&yy&i.&&dsname;
    tables &var/noprint out=freq&i;
    run;

   proc sql noprint;
    select nobs into :freqcnt from
    dictionary.tables
    where libname='WORK' & memname="FREQ&i";
    quit;

   %if &freqcnt <= 20 %then %do;
    proc freq data=in&i..&&mmm&i.&&y&i.&&dsname;
     tables &var/nopercent nocum missing;
     label &var="&&mmm&i.&&y&i &var";
     run;
```

```
%end;
%else %do;
 proc means
  data=in&i..&&mmm&i.&&y&i.&&dsname;
  var &var;
  label &var="&&mmm&i.&&y&i &var";
  run;
%end;
%mend in1;
```

CONCLUSION

This program demonstrates the use of SAS macros to compare frequencies and means of variables common to two data sets. It is useful for data review and comparisons to benchmark data, and can be adapted to other data comparison scenarios, such as comparison of more than two data sets or specifying different criteria for which variables are to be compared.

CONTACT INFORMATION

Aileen D. Bennett
U.S. Census Bureau
DSD, Room 1657, MS 8400
Washington, DC 20233
301-457-8052
Fax: 301-457-8077
aileen.d.bennett@census.gov