New SAS Tools For Database Management
N. D. Prabhakar
General Motors Corporation

INTRODUCTION

The power and versatility of SAS has made it an attractive package not only for purposes of statistical analysis but also for various data management and report generation functions. In fact we suspect that there is a sizable community of SAS users whose statistical needs do not extend beyond the procedures discussed in SAS User's Guide: Basics (1982). While SAS offers several general tools for routine manipulations with data sets, it would be convenient to have a set of pre-packaged macros and procedures to carry out various database management tasks with fewer and easily understandable instructions. In this paper, we describe a few such procedures and macros.

MACROS & PROCEDURES

(1) %MACRO CRDUNQ is a macro which creates an output SAS data set by taking unique combinations of values of specified variables from an input data set.

The macro uses the following four keyword parameters: DATA, OUT, BYSPEC, SORTED.

The value of the DATA should be the name of the input data set. Its default value is LAST. The value of OUT is the name of the output data set, with a default value of DATA.

The value of the parameter BYSPEC is the specification of all the BY variables, whose unique values will determine the observations of the output data set. The value of the parameter SORTED is used to determine whether or not the input data set is presorted according to the BYSPEC specification. If the value is Y or YES, the input data set is assumed to be presorted; otherwise, the macro will sort the data set according to the BYSPEC value. The default value of SORTED is N.

(2) %MACRO DUPS is a macro which creates a subset of an input SAS data set by taking only those observations in which a given set of variables have duplicate values in the input data set.

The macro has the same keyword parameters as %MACRO CRDUNQ with same meaning and default values.

(3) %MACRO NODUPS is a macro which creates a subset of an input SAS data set by taking only those observations in which a given set of variables do not have duplicate values in the input data set. The macro has the same keyword parameters as %MACRO DUPS with same meaning and default values.

(4) PROC PRODUCT is a procedure which creates an output SAS data set by combining each observation of one input data set with each observation of a second input data set.

The procedure is called by statements of the form:

PROC PRODUCT options;
BY variables;

The following options may be included in the PROC PRODUCT statement.

The options DATA_1 = SAS data set and DATA_2 = SAS data set specify the names of the input data sets used by the procedure. If either of these options is omitted, PRODUCT uses the last data set created as the corresponding input data set.

The option OUT = SAS data set specifies the name of the output data set to be created by PRODUCT. If the option is omitted, the procedure creates a new SAS data set and names it according to the DATA convention.

If a BY statement appears, then the output data set is created by combining each observation of one input data set within a BY group with each observation of the second input data set within the same BY group. If a BY statement appears, both input data sets are assumed to be sorted in order of the BY variables.

(5) %MACRO JOIN is a macro which creates a SAS data set by concatenating or interleaving a number of input data sets. It has four keyword parameters DATA, OUT, BYSPEC and SORTED. The value of DATA is the list of names of all input data sets. The other three parameters have the same meaning as in above macros. If the BYSPEC has a null value, then the input data sets are concatenated; otherwise they are interleaved using the variables in BYSPEC as BY variables. The default value for BYSPEC is null.
(6) %MACRO MEET is a macro which creates a SAS data set whose variables are specified variables common to a number of input data sets and whose observations are the values of the corresponding observations that occur in all the input data sets. It has four keyword parameters DATA 1, DATA 2, BYSPEC and SORTED, with same meaning as in %MACRO JOIN. The variables in the output data set will be the variables whose names appear in the value of BYSPEC.

(7) %MACRO DIFF is a macro which creates a SAS data set from two given input data sets such that the variables in the output data set are those occurring in the first input data set. The observations in the output data set are those observations of the first input data set such that values of specified common variables do not appear in the second input data set. The macro has five keyword parameters DATA 1, DATA 2, OUT, BYSPEC and SORTED. The value of DATA 1 is the name of the first input data set; The value DATA 2 is the name of the second input data set. The other three parameters have the same meaning as in %MACRO MEET.

(8) %MACRO INFER is a macro associated with PROC PRODUCT. It creates variables in a "source" data set by moving values of specified variables in a "target" data set using a given "relation" from the source to the target data set. It has seven keyword parameters SOURCE, TARGET, BYSPEC, VAR, NAME, SORTED and PRODUCT. SOURCE gives the name of the source data set. TARGET is the name of the target data set. BYSPEC gives the specification of the BY variables. The observations in the source data set whose BY values match with the BY values of observations in the target data set define the relation from the source data set to the target data set.

The parameter VAR gives the names of the variables in the target data set which will be moved over to the source data set. The parameter NAME gives the names of the variables created in the source data set corresponding to the variables listed in VAR. The parameter SORTED has the same meaning as in %MACRO JOIN. The parameter PRODUCT gives the name of the "product" data set of the source data set and the target data set using the given BY specification. If the value is null, the product data set will be created using PROC PRODUCT. The default value of PRODUCT is null value. The default values of SOURCE and TARGET are _LAST_.

Among the BY groups which occur in both input data sets, let the largest number of observations in the target data set appearing in a single BY group be M. Then for each variable X listed in VAR whose corresponding name in XX, the variables created in the source data set will be named XX1, XX2, ..., XXM. The value of XXI for any observation in the source data set will be the value of X for the I-th observation in the target data set which matches the source data set observation by having the same BY values. If a BY group in the source data set does not occur in the data set, then the newly created variables for the corresponding observations will be set to missing values.

(9) %MACRO SMZTHRU is a macro which creates variables in a source data set by computing sums, means, standard deviations, sum of squares, minima and maxima of specified variables for observations in a target data set which are "related" to the source data set observation by having the same BY values. The macro has seventeen keyword parameters SOURCE, TARGET, BYSPEC, SUMVAR, SUMNAME, MEANVAR, MEANNAME, SDVAR, SDNAME, SSDVAR, SSNAME, MINVAR, MINNAME, MAXVAR, MAXNAME, SORTED, PRODUCT.

The parameters SOURCE, TARGET, BYSPEC SORTED and PRODUCT have the same meaning and default values as in %MACRO INFER. The value of SUMVAR is a list of variables in the target data sets which are to be summed across the relation. The value of SUMNAME is a list of names for the variables created in the source data set corresponding to the variables listed in SUMVAR. For each variable in SUMVAR, a variable in the source data set will be created. Similarly MEANVAR, MEANNAME, SDVAR, SDNAME, SSDVAR, SSNAME, MINVAR, MINNAME, MAXVAR, MAXNAME, SORTED, PRODUCT.

CONCLUSION

In this paper we have described a few macros and procedures designed to carry out routine database management commands in a straightforward manner. A number of other macros have been developed by the author to (1) print out simple summary reports; (2) create variables and arrays in data sets; (3) create different subsets of data sets; (4) input data from OS data sets etc.
For details please contact:

General Motors Corporation
Dr. N.D. Prabhakar
485 W. Milwaukee
A-104D
Detroit, MI 48202

REFERENCES

SAS USER'S GUIDE: BASICS
1982 Edition
SAS Institute Inc.
Cary, N.C.

DATABASE DESIGN
G. Wiederhold
McGraw-Hill 1977

DATABASE SYSTEMS
J.D. Ullman
Computer Science Press
1980