

PO03

Selecting Variable Names into a Macro Variable Facilitates Data Analysis

Fenggang Peng, Nashville, TN

ABSTRACT

The Proc SQL select into is a powerful SAS tool. It can be used to generate one or more macro variables. The macro variable can include either all the values in a variable, all the variables in a dataset, or all the dataset names in a data library. The values in the macro variable can then be retrieved using %scan function, and referenced in title, footnote, data step, and procedure for data manipulation and data analysis. This paper will illustrate this technique by analyzing the real world data.

Keywords: INTO clause, macro, SQL, and %scan function

INTRODUCTION

For data analysis, we often need to make the same manipulation and analysis on all variables in a dataset or all datasets in a data library. Proc SQL select into technique provides an easy method to deal with such situations. In this paper we will briefly introduce this technique through analysis of the real word data.

DESCRIPTION OF DATA

In order to monitor the compliance of contracted providers with the clinical standard, the OMC clinical service staff in the Tennessee Department of Mental Health and Development Disabilities' (TDMHDD) Office of Managed Care (OMC) developed specific audit tools for each of particular provider service types. The service staff made visits to different provider sites to get the audit data. Among others, the audit data include provider's name, the consumer's SSN, and many questions to evaluate the compliance. The answers to these questions are either yes, no, unknown or N/A. The aim of the analysis is to generate frequencies of answer yes and no, and correspondent percentages to each question for each provider and each service type. For each question, only the yes and no answers are counted. For each service type, the analysis also generates the average of percentage of the answer yes for all questions.

PROCEDURE OF ANALYSIS

In order to meet above target, we import each dataset containing all the necessary variables (provider, SSN, and evaluating questions) to a SAS library:

```
libname ds 'C:\Projects\CLINICAL SUMMARY\2ndQ';
```

and write a macro program (%macro getresult) to analyze all the data (see appendix 1).

In the macro, firstly, all the variables are selected into a macro variable **varList**, and the number of variables into a macro variable **num** by using the proc SQL select into technique.

The select into technique can be used to generate one or more macro variables to store the values of one or more columns (variables). For example, we can generate a macro variable ssnlist to store all the SSN in the variable SSN in the dataset adultcm:

```
proc sql noprint;
  select ssn
    into :ssnlist separated by ' '
  from adultcm;
```

The generation of a macro variable to store all the variable names in a dataset or all the dataset names in a library is accomplished through a dictionary table.

'A dictionary table is a read-only SAS data view that contains information about SAS data libraries, SAS data sets, SAS macros, and external files that are in use or available in the current SAS session' (sas documentation). Each dictionary table has an associated PROC SQL view in the SASHELP library with a name starting with letter V. We can selectively read the view(s) in the SASHELP library for a specific purpose. For detailed introduction of SAS dictionary table and view, the readers are referred to papers by SAS institute (2005), Davis (2001) and Lafler (2005). In our situation, we need the view vcolumn, in which the column **name** contains all the variables in the current SAS

session. We use a WHERE clause to subset the view and generate a macro variable containing all the variable names in the dataset to be analyzed (step 1 in appendix 1).

Next, the data are manipulated to contain only yes, no, or missing value for the answers to each question. The manipulation is finished by a data step using an array. The elements in the array are contained in the macro variable varlist. Since the provider and SSN are the first two elements, the do loop begins at 3 (step 2 in appendix 1).

Finally, the percentages of yes and no in each question, and the average of all individual percentages are calculated. The calculations are done using proc freq and proc means. Each variable contained in the macro variable Varlist is reached through the %scan function and %do loop. Proc freq calculates the frequency and percentage, and outputs a SAS dataset for each value in the varlist. The outputted datasets are named answer, and combined together, from which the proc means calculates the average. The results of the two procedures are saved in the Microsoft Excel and Word, respectively, through ODS techniques (step3 in appendix1).

From the SAS codes, we can see that it is not necessary to know in advance in each dataset the variable names, which have been selected into a macro variable. Thus, we can write a reusable macro program to automate the data analysis.

CONCLUSION

Through generating a macro variable, we do not need to know in advance the values in a variable, the variable names in a dataset, or the dataset names in a library for data manipulation and analysis. This technique makes easy data manipulation and analysis.

REFERENCES

Michael Davis (2001), You Could Look It Up: An Introduction to SASHELP Dictionary Views, sugi 26 proceedings
Kirk Paul Lafler (2005), Exploring DICTIONARY Tables and Views, sugi 30 proceedings
SAS institute (2005), [SAS OnlineDoc 9.1.3 for the Web](http://support.sas.com/onlinedoc/913/docMainpage.jsp), <http://support.sas.com/onlinedoc/913/docMainpage.jsp>

ACKNOWLEDGMENTS

The author thanks bill polling who initiated the analysis of the project.

CONTACT INFORMATION

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

Fenggang Peng

Office of Managed Care

TN Department of Mental Health and Developmental Disabilities

425 5th ave., North

Nashville, TN 37243

Email: fpeng70802@yahoo.com

Fenggang.peng@state.tn.us

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

Appendix 1

```
dm editor "output; clear;log;clear" editor;
```

```
libname ds 'C:\Projects\CLINICAL SUMMARY\2ndQ';
```

```
%macro getresult(dataset=, provider=);
```

```
*****step 1. selection of variable names into a macro variable;
```

```
proc sql noprint;
    select name, count(name)
        into :varlist separated by ' ',
            :num
    from sashelp.vcolumn
    where libname='DS' and memname="&dataset";
quit;
```

```
*****step 2. transformation of answers in each question containing only Yes, No and missing value only;
```

```
data &dataset;
    set ds.&dataset;
    array fg[*] %str(&varlist);
    do i=3 to &num;
        if fg[i] in ('Yes','YES','yes') then
            fg[i]='Yes';
        else if fg[i] in ('NO', 'no', 'No') then
            fg[i]='No';
        else fg[i]=' ';
    end;
    drop i;
run;
```

```
*****step 3. calculation of percentages of Yes and No for each question and the average of all questions;
```

```
ods csv file = "C:\Projects\CLINICAL SUMMARY\2ndQ\&dataset.&provider..CSV";
```

```
%do i=3 %to &num;
    %let varname=%scan(&varlist,&i, ' ');
    proc freq data =&dataset;
        tables &varname/OUT=&varname;
        %if %upcase(&provider) ^=COMBINED %then
            %str(WHERE provider ="&provider");

    %if &i=3 %then
        %do;
            data answer;
            set &varname;
            rename &varname=category;
        %end;
    %if &i>3 %then
        %do;
            proc append base=answer
                data=&varname(rename=(&varname=category)) force;
        %end;
    %end;
ods csv close;
```

```
title1 ' The average of all questions ';
title2 "service Type=&dataset provider= &provider";
options nodate nonumber;
```

```

ods noproctitle;
ods rtf file= "C:\Projects\CLINICAL SUMMARY\2ndQ\&dataset.&provider..RTF";
proc means data=answer maxdec=2 noNobs;
  class category;
  var percent;
  label category=' ';
run;
ods rtf close;

*****deletion of the dataset answer;
proc datasets;
  delete ANSWER;
run;
quit;

%mend;

%getresult(dataset=CRG, provider=Foundations);
%getresult(dataset=CRG, provider=Pathways);
%getresult(dataset=CRG, provider=COMBINED);

```