ABSTRACT

The SAS® macro language is simple, yet powerful. List Processing with Proc SQL is also simple, yet powerful. This paper provides a data review macro FreqAll which illustrates using Proc SQL reading Dictionary.Columns to replace macro %do loops. Expected audience is advanced users, and macro programmers.

INTRODUCTION

Whenever I receive a data set, the first thing I want to do is examine the data, looking first at the data structure, (Proc Contents) then at a frequency listing of all the variables (Proc Freq). In data sets containing continuous variables the frequency listing gets long quickly. What I really want is similar to Proc Univariate: a list of the 10 high and low values. How can I make Proc Freq do that?

In this paper I develop a macro with a macro %do loop for each variable using Dictionary.Columns and then replace that loop with Proc SQL list processing.

The finished product is a listing which contains:

1. data structure list similar to Proc Contents
2. abbreviated frequency showing the high and low values
PROC CONTENTS

There are several ways to access the data structure of a data set:

1. Proc Contents:

```sas
Proc Contents data = SAShelp.Class;
```

2. Proc Datasets:

```sas
Proc Datasets library = SAShelp
details nolist
memtype = data;
contents
data = Class;
quit;
```

3. Proc Print:

```sas
PROC Print data = SAShelp.Vcolumn
(where = ( Libname eq 'SASHELP'
and MemName eq 'CLASS')
);
```

4. Proc SQL, describe:

```sas
PROC SQL; describe table SAShelp.Class
; quit;
```

5. Proc SQL, select:

```sas
PROC SQL; select Name, Type, Length, Label
from Dictionary.Columns
where Libname eq 'SASHELP'
and MemName eq 'CLASS'
; quit;
```

I work with the SQL-select example, which provides both the data structure list and, as I show later, the loop of macro calls.

PROC FREQ

Proc Frequency provides a listing of all values of a variable. For large data sets with continuous variables, the listing gets long quickly.

```sas
PROC Freq data = SAShelp.Prdsal2;
tables _all_
```

1800 lines = 36 pages!

<table>
<thead>
<tr>
<th>MONYR</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAN95</td>
<td>24</td>
<td>2.40</td>
<td>24</td>
<td>2.40</td>
</tr>
<tr>
<td>FEB95</td>
<td>24</td>
<td>2.40</td>
<td>48</td>
<td>4.80</td>
</tr>
</tbody>
</table>

My goal is an output of few pages: one for data structure listing and others for the frequency listing of high and low values.
PROC UNIVARIATE
Proc Univariate provides a listing of the extreme values of a variable, but only of the numerics.

```
PROC Univariate data = SAShelp.Class;
   var _numeric_;
```

The above examples illustrate my wish list: a list of variable attributes, and a limited frequency listing, showing only the extreme values.

In the next section I show a macro %do loop, illustrate how to use it for a procedure, and examine its programming issues.

MACRO %DO LOOP
A %do loop in a macro is similar to a data step loop. On listing line 28, log line 8, the index, I, is incremented from the lower bound, 1, to the upper bound, the macro variable Dim-Item. The macro variable array, Item, contains 3 elements, the sequentially numbered macro variables: Item1, Item2, and Item3. The dimension of the macro array is Dim-Item. This naming convention is necessary in order for the loop to access each element in the loop with the reference: double ampersand, array-name, index — &Item&I.— shown in log line 9.

```
1 %Macro Do_Loop;
2 %local Item1 Item2 Item3 Dim_Item I;
3 %let Item1 = X1 ;
4 %let Item2 = Y-2;
5 %let Item3 = Z 3;
6 %let Dim_Item = 3 ;
7 %Do I = 1 %to &Dim_Item.;
8 %Put Item&I<&&Item&I.>;
9 %end;
10 %mend;
11 %Do_Loop
12 Item1<X1>
13 Item2<Y-2>
14 Item3<Z 3>
```

The problems associated with using macro arrays are:

<table>
<thead>
<tr>
<th>log line</th>
<th>statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>%local Item1 ...</td>
</tr>
<tr>
<td>3-5</td>
<td>%Let Item? = ...</td>
</tr>
<tr>
<td>6</td>
<td>%Let Dim_Item = 3;</td>
</tr>
</tbody>
</table>
PROC SQL SELECT INTO

[Fehd:97a] illustrates using a Proc Contents output data set to create a macro array. [Fehd:04] shows how to use Proc SQL to do the same, as shown here. [Clay:06] provides comprehensive analysis of usage of macro arrays. [Whitlock:04] reviews macro design and list processing.

```
PROC SQL noprint;
  select Name
  into :Name1 - :Name999
  from Dictionary.Columns
  where Libname eq 'SASHELP'
    and MemName eq 'CLASS'
;quit;
%Put _User_;
```

Note the system-generated automatic macro variable SQLobs, listing line 35, has the upper bound value, 5.

MACRO FREQ-LOOP

In this section I show a demonstration macro which makes a macro array of the variables in a data set and then a macro %do loop.

As noted above, I use Dictionary.Columns to supply the listing of the data structure.

```
PROC SQL; describe table Dictionary.Columns;
quit;
```

This table contains these columns:

```
create table DICTIONARY.COLUMNS
  (  
    libname char(8) label='Library Name',
    memname char(32) label='Member Name',
    memtype char(8) label='Member Type',
    name char(32) label='Column Name',
    type char(4) label='Column Type',
    length num label='Column Length',
    npos num label='Column Position',
    varnum num label='Column Number in Table',
    label char(256) label='Column Label',
    format char(16) label='Column Format',
    informat char(16) label='Column Informat',
  )
```

The macro FreqAll in program FreqAll-Loop has parameters for: libref, data set name, how many extreme values to show, and testing (debugging).
Note that two macro arrays are created — Name and Type — in lines 28-29, and the scope of their variables is declared in lines 23-24.

The loop begins on line 37 and ends on line 54.

Note the four macro array element references (&&Var&I.) in lines:

39
41
42 (&&Type&I.)
46

For this demonstration I make a report with only the highest values. See the complete high and low processing in program FreqOf below.

FREQALL: SQL REPLACES MACRO DO LOOP

Where FreqAll was the name of the macro which contained the macro array and %do loop, here I name the program FreqAll and have placed the statements inside the %do loop into the macro subroutine FreqOf.
Note that the parameter names are aligned with the variables from the Dictionary.Columns data set: Name, Type, Length, Format and Label.

If the user desires not the highest and lowest values but the highest and lowest frequencies, then I have provided a parameter, Order = freq (lines 50: default, 52: must be enabled by opening comment on line 49 and closing comment on line 51), which shows the mode: the values occurring most and least often.

Note that the parameters InLib, InData, and Nobs2View refer to global macro variables set before the macro is called.

```sas
%Macro Freqof
  (Name = /* var */
   ,Type = /* in (char,num) */
   ,Length = /* integer */
   ,Format = /* $char */
   ,Label = /* $char40 */
   /* for hi and low, use either: ****** */
   ,Order = internal /* default: hi and low values */
   /* for mode use: ** *** *** ****** ** */
   ,Order = freq /* hi and low descending count */
   ,InLib = &In_Lib. /* FreqAll scope: global */
   ,InData = &In_Data. /* FreqAll scope: global */
   ,Nobs2View = &HiLowToView /* FreqAll scope: global */
   ,Testing = 0 /* show stuff? ,Testing=1 */
  )/des = 'site: FreqOf: subroutine of prog FreqAll'
  /*change notes
```
Compare to program Freq-All-Loop.

The Proc Freq is the same except for the addition of the order parameter. The macro array references (&Name&I., &Type&I.) have been changed to parameter name (macro variable) references: &Name, &Type.

Data Freq, attrub standardizes the data set structure.

```
%local Nobs; %let Nobs = 0;%*initialize for symput;
PROC Freq data = &InLib..&InData.
   order = &Order.;
   tables &Name
      / list missing noprint
      out = Freq(rename = (&Name. =
         %If &Type = char %then ValuC;
         %else ValuN;
      ));
DATA Freq;
   attrib Attributes length = %eval(32 +1 +4 +1 +4 +1 + 40)
      ValuC length = $20
      ValuN length = 8
      Count length = 4 format = comma.
      Percent length = 8 format = 6.2
      Level length = 4;
   retain Attributes "&Name. &Type.:&Length. &Label"
      ValuC "." ValuN Level .;
   do until(EndoFile);
      set Freq end = EndoFile
         nobs = Nobs;
      Level ++1; %*increment retained counter;
      output; end;
   stop; run;
%If &Testing. %then %Put Note:&SysMacroName.: nobs<&Nobs.>
%If &Nobs. le %eval(2 * &Nobs2View. + 2) %then %do;
      PROC Append base = ListFreq
         data = Freq(obs = &Nobs2View.);
      data = Freq(firstobs = %eval( &Nobs.
         - &Nobs2View. +1));
%end;%*Else do ......................... *
run;%* ................................. *
%Mend Freqof;
```

This section either appends a small listing
or divides the list into high and low sets
of values and adds a note indicating that
values were removed.
In order for the global macro variables
In_Lib and In_Data

FREQAll.sas program FreqOf snip

```sas
%Let In_Lib = Library;
%Let In_Data = MyData ;
%Let HiLowToView = 5 ;
```
to be used in the `select ... from ... where phrase`, lines 128-129 and 133-134, each must be in ALL CAPS.

Proc SQL creates three objects:

1. line 124: table `ListAttributes` containing the variable attributes; this is the first page of the summary report.

2. line 137: macro variable `List` containing calls of macro `FreqOf` for each variable; these are executed on line 145.
   To view the `FreqOf` statements, disable line 121:
   ```sas
   %*Let SQLprint = noprint;
   This select statement, lines 131-138, replaces the macro array and %do loop in the FreqAll-Loop program. Note: a macro variable for the upper bound is not needed.
   ```
3. line 139: macro variable `NobsData`: the number of observations of the input data set; this is used in the title2 statement, lines 148-149.

The report is printed in two parts: attributes, and frequencies.

Housecleaning: delete the program’s global macro variables.

```sas
%Let In_Lib = %upcase(&In_Lib.);
%Let In_Data = %upcase(&In_Data.);
%Let SQLprint = print;%*testing: view FreqOf statements;
%Let SQLprint = noprint;
PROC SQL &SQLprint.;
   create table ListAttributes as
   select Varnum, Name, Type, Length, Format, Informat, Label, Npos
   from Dictionary.Columns
   where Libname eq "&In_Lib."
   and MemName eq "&In_Data."
   and MemType eq 'DATA'
   ; select '%FreqOf(name = ' !! trim(Name) !! ',type = ' !! trim(Type) !! ',length = ' !! compress(put(Length,32.)) !! ',label = ' !! trim(Label) !! ')'
   into :List separated by ' '
   from ListAttributes
   ; select Nobs into :NobsData
   from Dictionary.Tables
   where Libname eq "&In_Lib."
   and MemName eq "&In_Data."
   and MemType eq 'DATA'
   ; quit;
%List. ; %*execute macro calls;
%Let NobsData = &NobsData;%* remove leading blanks;
Title2 "&SysProcessName.: &In_Lib..&In_Data. nobs:&NobsData."
PROC Print data = ListAttributes noobs;
   Title3 "variable attributes";
PROC Print data = ListFreq;
   var Valu: Count Percent Level;
   by Attributes notsorted;
   id Attributes ;
   Title3 "list of variable frequencies";
run;
%symdel In_Data In_Lib HiLowToView List NobsData SQLprint;
```
Changing line 121 to:

```sql
/* Let SQLprint = noprint;
 produces this output, which shows the
 statements in the macro variable List.
 Note: spaces have been added to align
 columns and improve readability.
```

This is the FreqAll report for SAShelp.PrdSal2; compare to program F-Freq.sas.

The first page of the FreqAll report contains Proc Contents information.

The second page contains the abbreviated frequencies of each variable.

Note: the listing is truncated to save space.

The complete listing from program FreqAll of SAShelp.PrdSal2 is approximately 100 lines; three pages, instead of 20 from Proc Contents: 2 pages
Proc Freq: 18 pages

<table>
<thead>
<tr>
<th>varnum</th>
<th>name</th>
<th>type</th>
<th>length</th>
<th>format</th>
<th>informat</th>
<th>label</th>
<th>npos</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>COUNTRY</td>
<td>char</td>
<td>10</td>
<td>$CHAR10.</td>
<td></td>
<td>Country</td>
<td>48</td>
</tr>
<tr>
<td>2</td>
<td>STATE</td>
<td>char</td>
<td>22</td>
<td>$CHAR22.</td>
<td></td>
<td>State/Province</td>
<td>58</td>
</tr>
<tr>
<td>3</td>
<td>COUNTY</td>
<td>char</td>
<td>20</td>
<td>$CHAR20.</td>
<td></td>
<td>County</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>ACTUAL</td>
<td>num</td>
<td>8</td>
<td>DOLLAR12.2</td>
<td></td>
<td>Actual Sales</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>PREDICT</td>
<td>num</td>
<td>8</td>
<td>DOLLAR12.2</td>
<td></td>
<td>Predicted Sales</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attributes</th>
<th>ValuC</th>
<th>ValuN</th>
<th>Count</th>
<th>Percent</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNTRY char:10 Country</td>
<td>U.S.A.</td>
<td>1,000</td>
<td>100.00</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>STATE char:22 State/Province California</td>
<td>1,000</td>
<td>100.00</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COUNTY char:20 County</td>
<td>.</td>
<td>1,000</td>
<td>100.00</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ACTUAL num:8 Actual Sales</td>
<td>.</td>
<td>8.16</td>
<td>0.10</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>10.20</td>
<td>0.10</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>13.60</td>
<td>0.10</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>17.00</td>
<td>0.10</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>27.20</td>
<td>0.10</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;snipped&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>3410.20</td>
<td>0.10</td>
<td>851</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>3413.60</td>
<td>0.10</td>
<td>852</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>3415.30</td>
<td>0.10</td>
<td>853</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>3440.80</td>
<td>0.10</td>
<td>854</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>3515.60</td>
<td>0.10</td>
<td>855</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PREDICT num:8 Predicted Sales</td>
<td>.</td>
<td>4.08</td>
<td>0.20</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
[Fehd:05b] discusses necessary items in a program header.

To receive the latest edition of this program send an e-mail to the author [mailto:RJF2@cdc.gov] with the subject: request FreqAll

---

Suggested Readings

- [Abolafia:97] provides a macro to reproduce Proc DataChk.
- [Carpenter:2004, ch. 9] discusses dynamic programming (list processing).

CONCLUSION

**FreqAll** The data review utility program FreqAll provides a shorter data set summary with more information.

**Proc SQL** List processing (select ... into :List) can eliminate the use of macro arrays and %do loops. This yields clearer code.
Acknowledgements

My colleagues at CDC, too many to mention here, provided the dirty data for which I originally developed this routine in the early 1990s. Toby Dunn provided commentary and critique. Dianne Rhodes whispered SQL encouragement to me. I am grateful to Ian Whitlock and for his many contributions to SAS-L, the on-line SAS User Group; he raised the bar.

REFERENCES

http://www2.sas.com/proceedings/sugi22/POSTERS/PAPER229.PDF

http://support.sas.com/publishing/bbu/companion_site/59224.html


http://www2.sas.com/proceedings/sugi22/CODERS/PAPER80.PDF

http://www2.sas.com/proceedings/sugi29/070-29.pdf

http://www2.sas.com/proceedings/sugi30/067-30.pdf

http://www2.sas.com/proceedings/sugi29/244-29.pdf

http://www2.sas.com/proceedings/sugi22/ADVTUTOR/PAPER42.PDF

Author: Ronald Fehd mailto:Ronald.Fehd@cdc.hhs.gov
Centers for Disease Control MS-G23 e-mail: RJF2@cdc.gov
4770 Buford Hwy NE
Atlanta GA 30341-3724 bus: 770/488-8221

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.

about the author:

education: B.S. Computer Science, U/Hawaii, 1986
SUGI attendee since 1989
SAS-L reader since 1994

experience: programmer: 20+ years
data manager at CDC, using SAS: 18+ years
author: 10+ SUG papers
author: 3,000+ messages to SAS-L since 1997

SAS-L:

http://www.listserv.uga.edu/cgi-bin/wa?S1=sas-l
Search for:
The subject is or contains: LaTeX
The author’s address: RJF2
Since: 01 June 2003

Document Production: This paper was typeset in LaTeX. For further information about using LaTeX to write your SUG paper, consult the SAS-L archives: