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.

Keywords: data review, dynamic programming, list processing, do loop, macro, SQL.

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

Contents

| Proc Contents | 2 |
| Proc Freq | 2 |
| Proc Univariate | 3 |
| Macro %Do Loop | 3 |
| Proc SQL select into | 4 |
| Macro Freq-Loop | 4 |
| FreqAll: SQL replaces macro %do loop | 5 |
| Conclusion | 8 |
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.

```
1  PROC Freq data = SAShelp.Prdsal2;
2    tables _all_;
```

1800 lines ≈ 36 pages!

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

```sas
PROC Univariate data = SAShelp.Class;
  var _numeric_;
F-Univariate.sas
```

The UNIVARIATE Procedure

Variable: Age

### Extreme Observations

<table>
<thead>
<tr>
<th>Value</th>
<th>Obs</th>
<th>Value</th>
<th>Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>18</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>15</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>15</td>
<td>19</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>16</td>
<td>15</td>
</tr>
</tbody>
</table>

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.

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>

To ensure the scope of macro array variables, allocation of each element, and allocation of the dimension (upper bound),
**PROC SQL SELECT INTO**


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.

This table contains these columns:

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.
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, attrib standardizes the data set structure.

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 \texttt{In\_Lib} and \texttt{In\_Data} to be used in the \texttt{select} ... \texttt{from} ... \texttt{where} phrase, lines 128-129 and 133-134, each must be in ALL CAPS.

Proc SQL creates three objects:

1. line 124: table \texttt{ListAttributes} containing the variable attributes; this is the first page of the summary report.

2. line 137: macro variable \texttt{List} containing calls of macro \texttt{FreqOf} for each variable; these are executed on line 145.

To view the \texttt{FreqOf} statements, disable line 121:
\begin{verbatim}
%*Let SQLprint = noprint;
\end{verbatim}

This select statement, lines 131-138, replaces the macro array and \%do loop in the \texttt{FreqAll-Loop} program. Note: a macro variable for the upper bound is not needed.

3. line 139: macro variable \texttt{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.

Changing line 121 to:
\begin{verbatim}
%Let SQLprint = noprint;
\end{verbatim}

produces this output, which shows the statements in the macro variable \texttt{List}. Note: spaces have been added to align columns and improve readability.

This is the \texttt{FreqAll} report for \texttt{SAShelp.Prdsal2}; compare to program \texttt{F-Freq.sas}.

The first page of the \texttt{FreqAll} report contains Proc \texttt{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

FreqAll

Fehd in [Prog-Doc] 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[Data-Check] provides a macro to reproduce Proc DataChk.
- Wobus and Gober[Data-Analysis] show data review with procs Summary and Univariate.

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
4770 Buford Hwy NE
Atlanta GA 30341-3724
bus: 770/488-8221
e-mail: RJF2@cdc.gov

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 since1997

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.

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

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