

# Journeyman's Tools: Data Review Macro FreqAll: Using Proc SQL List Processing with Dictionary.Columns to Eliminate Macro Do Loops

Ronald J. Fehd, Centers for Disease Control and Prevention, Atlanta, GA, USA

## 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 will 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 will be a listing which contains:

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

## Contents

|                                              |          |
|----------------------------------------------|----------|
| <b>Proc Contents</b>                         | <b>2</b> |
| 1.1 C-Contents.sas . . . . .                 | 2        |
| 1.2 C-Datasets.sas . . . . .                 | 2        |
| 1.3 C-Print.sas . . . . .                    | 2        |
| 1.4 C-SQL-describe.sas . . . . .             | 2        |
| 1.5 C-SQL-select.sas . . . . .               | 2        |
| <b>Proc Freq</b>                             | <b>2</b> |
| 2.1 F-Freq.sas . . . . .                     | 2        |
| 2.2 F-Freq.lst . . . . .                     | 2        |
| <b>Proc Univariate</b>                       | <b>3</b> |
| 3.1 F-Univariate.sas . . . . .               | 3        |
| 3.2 F-Univariate.lst snip 1 . . . . .        | 3        |
| 3.3 F-Univariate.lst snip 2 . . . . .        | 3        |
| <b>Macro Do Loop</b>                         | <b>3</b> |
| 4.1 Macro-do-loop.log . . . . .              | 3        |
| <b>Proc SQL select into</b>                  | <b>3</b> |
| 5.1 SQL-select-into.sas . . . . .            | 3        |
| 5.2 SQL-select-into.log . . . . .            | 4        |
| <b>Macro Freq-Loop</b>                       | <b>4</b> |
| 6.1 Describe-Dict-Columns.sas . . . . .      | 4        |
| 6.2 Describe-Dict-Columns.log . . . . .      | 4        |
| 6.3 FreqAll-Loop.sas snip 1 . . . . .        | 4        |
| 6.4 FreqAll-Loop.sas snip 2 . . . . .        | 4        |
| 6.5 FreqAll-Loop.sas snip 3 . . . . .        | 5        |
| <b>FreqAll: SQL replaces macro do loop</b>   | <b>5</b> |
| 7.1 FreqAll.sas macro FreqOf pg. 1 . . . . . | 5        |
| 7.2 FreqAll.sas macro FreqOf pg. 2 . . . . . | 6        |
| 7.3 FreqAll.sas List Processing . . . . .    | 7        |
| 7.4 FreqAll-List-of-FreqOf.txt . . . . .     | 7        |
| 7.5 FreqAll.txt PrdSal2 pg. 1 . . . . .      | 7        |
| 7.6 FreqAll.txt PrdSal2 pg. 2 . . . . .      | 8        |
| 7.7 FreqAll.sas documentation . . . . .      | 8        |
| <b>Suggested Readings</b>                    | <b>8</b> |
| <b>Conclusion</b>                            | <b>9</b> |

# PROC CONTENTS

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

1. Proc Contents:

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

2. Proc Datasets:

```
1 Proc Datasets library = SAShelp
2 details nolist
3 memtype = data;
4 contents
5 data = Class;
6 quit;
```

3. Proc Print:

```
1 PROC Print data = SAShelp.Vcolumn
2 (where = ( Libname eq 'SASHELP'
3 and MemName eq 'CLASS'
4 ) );
```

4. Proc SQL, describe:

```
1 PROC SQL; describe table SAShelp.Class
2 ; quit;
```

5. Proc SQL, select:

```
1 PROC SQL; select Name, Type, Length, Label
2 from Dictionary.Columns
3 where Libname eq 'SASHELP'
4 and MemName eq 'CLASS'
5 ; quit;
```

I will be working with the SQL-select example, which will provide both the data structure list and 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!

| F-Freq.lst |           |         |                      |                    |
|------------|-----------|---------|----------------------|--------------------|
| Month/Year |           |         |                      |                    |
| MONYR      | Frequency | Percent | Cumulative Frequency | Cumulative Percent |
| JAN95      | 24        | 2.40    | 24                   | 2.40               |
| FEB95      | 24        | 2.40    | 48                   | 4.80               |

My goal is an output of few pages: one for data structure listing and others for frequency listing.

# PROC UNIVARIATE

Proc Univariate provides a listing of the extreme values of a variable, but only numerics.

1

2

----- F-Univariate.sas -----  
PROC Univariate data = SAShelp.Class;  
var \_numeric\_;

3

4

----- F-Univariate.lst snip 1 -----  
The UNIVARIATE Procedure  
Variable: Age

52

53

54

55

56

57

58

59

60

61

62

...  
----- F-Univariate.lst snip 2 -----  
Extreme Observations  
  
----Lowest----      ----Highest---  
  
Value      Obs      Value      Obs  
  
11      18      15      8  
11      11      15      14  
12      16      15      17  
12      13      15      19  
12      10      16      15

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 will show a macro do loop, illustrate how to use it for a procedure, and examine its programming problems.

# 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.

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

----- Macro-do-loop.log -----  
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  
8      %do I = 1 %to &Dim\_Item.;  
9          %put Item&I<&&Item&I.>;  
10      %end;  
11      %mend;  
12      %Do\_Loop  
Item1<X1>  
Item2<Y-2>  
Item3<Z 3>

The problems associated with using macro arrays are:

|                                           | log line | statement          |
|-------------------------------------------|----------|--------------------|
| ensure the scope of macro array variables | 2        | %local Item1 ...   |
| allocation of each element                | 3-5      | %let Item? = ...   |
| allocation of the dimension (upper bound) | 6        | %let Dim.Item = 3; |

# PROC SQL SELECT INTO

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

1

2

3

4

5

6

7

8

----- SQL-select-into.sas -----  
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.

```

35 GLOBAL SQLobs 5
36 GLOBAL SQLoops 38
37 GLOBAL NAME4 Height
38 GLOBAL NAME5 Weight
39 GLOBAL NAME2 Sex
40 GLOBAL NAME3 Age
41 GLOBAL NAME1 Name

```

## MACRO FREQ-LOOP

In this section I will 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 will use `Dictionary.Columns` to supply the listing of the data structure.

```

1 PROC SQL; describe table Dictionary.Columns;
2 quit;

```

This table contains these columns:

```

28 libname char(8) label='Library Name',
29 memname char(32) label='Member Name',
30 memtype char(8) label='Member Type',
31 name char(32) label='Column Name',
32 type char(4) label='Column Type',
33 length num label='Column Length',
34 npos num label='Column Position',
35 varnum num label='Column Number in Table',
36 label char(256) label='Column Label',
37 format char(16) label='Column Format',
38 informat char(16) label='Column Informat',

```

Macro `FreqAll` has three parameters: `libref`, data set name and how many extreme values to show.

```

4 %Macro FreqAll /* ----- */
5 (In_Lib = SASHELP
6 ,In_Data = Class
7 ,ShowHiLo = 5

```

Note that two macro arrays are created — `Name` and `Type` — in lines 29-30, and the scope of their variables is declared in lines 25-26.

```

25 %** Initialize macro array; %local SqlObs SqlLoops;
26 %Do I = 1 %to 999; %local Var&I. Type&I.; %end;
27
28 PROC SQL noprint;
29 select Name, Type
30 into :Var1 - :Var999
31 ,:Type1 - :Type999
32 from Dictionary.Columns
33 where LibName eq "&In_Lib."
34 and MemName eq "&In_Data."
35 ;quit;

```

The loop begins on line 39 and ends on line 56.

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

```
41
43
44 (&Type&I.)
48
```

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

```

                                FreqAll-Loop.sas snip 3
39 %Do I = 1 %to &SQLobs.;%* ----- *;
40 Proc Freq data = &In_Lib..&In_Data.;
41     tables    &&Var&I.
42             / list missing noprint
43             out = Freq(rename = (&&Var&I. =
44                             %If &&Type&I. eq char %then ValuC;
45                             %Else                               ValuN;
46                             )                               );
47 DATA    Freq;
48 retain Var "&&Var&I.";
49 if      0 then          %** copy structure;
50 set      FreqList;
51 set      Freq;
52
53 PROC Append base = FreqList
54             data = Freq(obs = &ShowHiLo.)
55             force;
56 %end;%do I ..... *;
57
58 PROC Print data = FreqList;
59         by      Var notsorted;
60         id       Var;
61 run;%* ..... *;%Mend;

```

## 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, which, will show the mode: the values occurring most and least often.

Note that the parameters InLib, InData, and Nobs2View refer to global macro variables.

```

                                FreqAll.sas macro FreqOf pg. 1
43 %Macro Freqof
44 (Name      = /* var                                */
45 ,Type      = /* in (char,num)                      */
46 ,Length    = /* integer                            */
47 ,Format    = /* $char                               */
48 ,Label     = /* $char40                             */
49           /* for hi and low, use either:  ***** */
50 ,Order      = internal /* default: hi and low values */
51           /* for mode      use:  ** *** ** *  ***** **
52 ,Order      = freq    /* hi and low descending count */
53 ,InLib      = &In_Lib.    /* FreqAll scope: global */
54 ,InData     = &In_Data.   /* FreqAll scope: global */
55 ,Nobs2View  = &HiLowToView /* FreqAll scope: global */
56 ,Testing    = 0          /* show stuff? ,Testing=1 */
57 )/des       = 'site: FreqOf: subroutine of prog FreqAll'
58 ;/*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, `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.

```

65 _____ FreqAll.sas macro FreqOf pg. 2 _____
66 %local Nobs; %let Nobs = 0; %*initialize for symput;
67
68 PROC Freq data = &InLib..&InData.
69     order = &Order.;
70     tables &Name
71     / list missing noprint
72     out = Freq(rename = (&Name. =
73         %If &Type = char %then ValuC;
74         %else ValuN; ));
75
76 DATA Freq;
77     attrib Attributes length = %eval(32 + 1 + 4 + 1 + 4 + 1 + 40)
78     ValuC length = $20
79     ValuN length = 8
80     Count length = 4 format = comma.
81     Percent length = 8 format = 6.2
82     Level length = 4;
83
84 retain Attributes "&Name. &Type.:&Length. &Label"
85     ValuC "." ValuN Level .;
86
87 do until(EndoFile);
88 set Freq end = EndoFile
89     nobs = Nobs;
90 Level ++1; %*increment retained counter;
91 output; end; %*do until EndoFile;
92 call symput( 'Nobs', compress(put(Nobs,32.)));
93 stop; run;
94 %If &Testing. %then %Put Note:&SysMacroName.: nobs<&Nobs.>;
95
96 %If &Nobs. le %eval(2 * &Nobs2View. + 2) %then %do;
97 PROC Append base = ListFreq
98     data = Freq; %end;
99
100 %Else %do; %* ----- *;
101 DATA Snipped;
102 set Freq(obs = 1); ValuC = '<snipped>'; ValuN = .;
103 Count = .; Percent = .;
104 Level = .; output; stop;
105
106 PROC Append base = ListFreq %*high values;
107     data = Freq(obs = &Nobs2View.);
108
109 PROC Append base = ListFreq
110     data = Snipped;
111
112 PROC Append base = ListFreq %*low values;
113     data = Freq(firstobs = %eval( &Nobs.
114         - &Nobs2View. + 1));
115 %end; %*Else do ..... *;
116 run; %* ..... *; %Mend Freqof;

```

In order for the global macro variables `In_Lib` and `In_Data` 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:  
`%*Let SQLprint = noprint;`  
This `select` statement, lines 131-138, replaces the macro array and `do loop` in the `FreqAll-Loop` program.
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.

Changing line 121 to:

`%*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.

```
117 FreqAll.sas List Processing
118 %Let In_Lib = %upcase(&In_Lib.);
119 %Let In_Data = %upcase(&In_Data.);
120
121 %Let SQLprint = print; %*testing: view FreqOf statements;
122 %Let SQLprint = noprint;
123
124 PROC SQL &SQLprint.;
125     create table ListAttributes as
126     select Varnum, Name, Type, Length
127            , Format, Informat, Label, Npos
128            from Dictionary.Columns
129            where Libname eq "&In_Lib."
130                  and MemName eq "&In_Data."
131                  and MemType eq 'DATA'
132    ; select '%FreqOf(name = ' !! trim(Name)
133            !! ',type = ' !! trim(Type)
134            !! ',length = '
135            !! compress(put(Length,32.))
136            !! '),'
137    into :List separated by ' '
138    from ListAttributes
139    ; select Nobs into :NobsData
140    from Dictionary.Tables
141    where Libname eq "&In_Lib."
142          and MemName eq "&In_Data."
143          and MemType eq 'DATA'
144    ; quit;
145 &List. ; %*execute macro calls;
146
147 %Let NobsData = &NobsData; %* remove leading blanks;
148 Title2
149     "&SysProcessName.: &In_Lib..&In_Data. nobs:&NobsData.";
150
151 PROC Print data = ListAttributes noobs;
152     Title3 "variable attributes";
153
154 PROC Print data = ListFreq;
155     var Valu: Count Percent Level;
156     by Attributes notsorted;
157     id Attributes ;
158     Title3 "list of variable frequencies";
159 run;
160 %symdel In_Data In_Lib HiLowToView List NobsData SQLprint;
```

```
1 FreqAll-List-of-FreqOf.txt
2 Program FreqAll: SASHELP.CLASS nobs:19
3 %FreqOf(name = Name , type = char, length = 8, label = )
4 %FreqOf(name = Sex , type = char, length = 1, label = )
5 %FreqOf(name = Age , type = num , length = 8, label = )
6 %FreqOf(name = Height, type = num , length = 8, label = )
7 %FreqOf(name = Weight, type = num , length = 8, label = )
```

|                           |                                               |          |      |        |            |          |                 |      |
|---------------------------|-----------------------------------------------|----------|------|--------|------------|----------|-----------------|------|
| FreqAll.txt PrdSal2 pg. 1 |                                               |          |      |        |            |          |                 |      |
| 1                         | SUG-2006: data review utility program FreqAll |          |      |        |            |          |                 |      |
| 2                         | Program FreqAll: SASHELP.PRDSAL2 nobs:23040   |          |      |        |            |          |                 |      |
| 3                         | variable attributes                           |          |      |        |            |          |                 |      |
| 4                         |                                               |          |      |        |            |          |                 |      |
| 5                         | varnum                                        | name     | type | length | format     | informat | label           | npos |
| 6                         |                                               |          |      |        |            |          |                 |      |
| 7                         | 1                                             | COUNTRY  | char | 10     | \$CHAR10.  |          | Country         | 48   |
| 8                         | 2                                             | STATE    | char | 22     | \$CHAR22.  |          | State/Province  | 58   |
| 9                         | 3                                             | COUNTY   | char | 20     | \$CHAR20.  |          | County          | 80   |
| 10                        | 4                                             | ACTUAL   | num  | 8      | DOLLAR12.2 |          | Actual Sales    | 0    |
| 11                        | 5                                             | PREDICT  | num  | 8      | DOLLAR12.2 |          | Predicted Sales | 8    |
| 12                        | 6                                             | PRODTYPE | char | 10     | \$CHAR10.  |          | Product Type    | 100  |
| 13                        | 7                                             | PRODUCT  | char | 10     | \$CHAR10.  |          | Product         | 110  |
| 14                        | 8                                             | YEAR     | num  | 8      | 4.         |          | Year            | 16   |
| 15                        | 9                                             | QUARTER  | num  | 8      | 8.         |          | Quarter         | 24   |
| 16                        | 10                                            | MONTH    | num  | 8      | MONNAME3.  |          | Month           | 32   |
| 17                        | 11                                            | MONYR    | num  | 8      | MONYY.     | MONYY.   | Month/Year      | 40   |

Proc Freq: 18 pages

21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53

```

1      /*      Name: FreqAll.sas
2
3  Requirements   : description   : Proc Freq of all vars in data set
4                  purpose       : provide shorter listing than tables _all_
5
6  -----
7  Contexts      : program group: data review
8                  program  type: routine
9                  SAS       type: program with parameters
10                 uses routines: in-program macro %FreqOf
11
12  -----
13  Specifications: input   : libref
14                        data
15                        max number of high and low obs to view
16                  process: SQL   writes macro calls
17                        Proc Freq of each var
18                        save to data
19                        append to report
20                        subset if Nobs greater than Max-N-to-view
21                  output  : print report
22
23  -----
24  Information   :   author: Ronald J. Fehd
25
26  -----
27  Usage Examples:
28  %Let In_Lib   = Library;
29  %Let In_Data  = MyData ;
30  %Let HiLowToView = 5      ;
31
32  %Inc 'FreqAll.sas';

```

## SUGGESTED READINGS









First Contents Prev Next Appendix Last Quit

## 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.

## REFERENCES

- [Data-Check] Jeffrey M. Abolafia, *PROC DATACHK Revisited: The DATACHK Macro*, Proceedings of the 22nd Annual SAS® Users Group International Conference, 1997.  
<http://www2.sas.com/proceedings/sugi22/POSTERS/PAPER229.PDF>
- [Tight-Looping] Ted Clay, *Tight Looping With Macro Arrays*, Proceedings of the 31st Annual SAS® Users Group International Conference, 2006.  
<http://www2.sas.com/proceedings/sugi31/040-31.pdf>
- [Array-Contents] Ronald Fehd, *%Array: construction and usage of arrays of macro variables*, Proceedings of the 22nd Annual SAS® Users Group International Conference, 1997.  
<http://www2.sas.com/proceedings/sugi22/CODERS/PAPER80.PDF>
- [Array-SQL] Ronald Fehd, *%Array: construction and usage of arrays of macro variables*, Proceedings of the 29th Annual SAS® Users Group International Conference, 2004.  
<http://www2.sas.com/proceedings/sugi29/070-29.pdf>
- [Prog-Doc] Ronald Fehd, *Journeyman's Tools: The Writing For Reading And Reuse Program Header*, Proceedings of the 30th Annual SAS® Users Group International Conference, 2005.  
<http://www2.sas.com/proceedings/sugi30/067-30.pdf>
- [Macro-Design] Ian Whitlock, *A Second Look at SAS® Macro Design Issues*, Proceedings of the 29th Annual Conference of the SAS® Users Group, Int'l, 2004.  
<http://www2.sas.com/proceedings/sugi29/244-29.pdf>
- [Data-Analysis] Diana Zhang Wobus and John C. Gober, *A Step-by-Step Illustration of Building a Data Analysis Tool with Macros*, Proceedings of the 22nd Annual SAS® Users Group International Conference, 1997.  
<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**

**Document Production:** This paper was typeset in L<sup>A</sup>T<sub>E</sub>X. For further information about using L<sup>A</sup>T<sub>E</sub>X 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

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                    |

---

|        |                                             |
|--------|---------------------------------------------|
| SAS-L: | author: 3,000+ messages to SAS-L since 1997 |
|        | Most Valuable SAS-L contributor: 2001, 2003 |

---