

VRANGE: Deriving A Minimal FORMAT from a Numeric Variable's Range of Values

Andy Barnett, Clinical Trials Support Services, Burlington, NC

VRANGE is a SAS[®] macro that suggests a "minimal FORMAT" for each user-selected variable of a SAS dataset. The minimal FORMAT allows the data to be displayed in the smallest width with the largest precision that the data appears to use. These formats may then be used for such tasks as reporting data as compactly as possible without arbitrary precision in the form of excessive trailing zeros, or to produce listings with the decimal point in the same column of every row of a column of numbers (Goldstein, 1992, and Barlow and Ringelberg, 1992). For example, if a dataset contained laboratory results of unknown precision, VRANGE could suggest a precision by assuming that (1) the lab results were obtained in base-10 form, and (2) it is unlikely that all values of the variable have significant trailing zeros.

The SAS format, BESTw. (Table 13.3, p. 427, SAS Institute Inc., 1988) performs this task, but it does so on a value-by-value basis, rather than variable-by-variable. For example, if a variable has two values, say 1012.3 and 1.0123, and they are printed with format BEST6., then the decimal points will not line up. VRANGE uses the results of BEST32. (the maximum allowed), and determines a minimal format based on the maximum numbers found. This is shown below.

BEST6.	VRANGE=9.4
1012.3	1012.3000
1.0123	1.0123

Thus, an algorithm like VRANGE is needed to create listings with a single format to show all useful decimal places of a variable. Furthermore, it is likely that several variables from a dataset need to be handled in this manner, so a macro that would process several variables from a dataset might be more useful. VRANGE solves these problems.

The code for VRANGE is shown in Appendix A. It was produced under SAS 6.04 using MS-DOS[®] 5.0. As can be seen, there are four macro parameters. The first, INSSD, is the input dataset name containing the variable(s); the second is OUTSSD, the output dataset name for the formats; the third, VARS, is a list of the variable name(s); and the fourth is MINIMAL, an

optional output dataset that contains all the data of INSSD with the new formats. In addition to the code, a log and print file are listed which show a sample of VRANGE's output.

Examination of the VRANGE macro shows that for each variable specified in VARS, five variables are created by VRANGE; these are elements of the arrays VARS, CVARS, IVARS, FVARS, and FMT. VARS contains the INSSD VARS data, CVARS is a BEST32 copy of VARS, IVARS and FVARS are the maximum number of places before and after the decimal point in CVARS, respectively, and FMT is the final w.d format for each VARS. All that is happening is that the VARS are being converted to true decimal representations by converting them to characters with BEST32, and these decimal representations are then used to determine precision.

If all values of a variable are missing, then IVARS and FVARS are both zero. FMT will then be "1.0", thus providing one position to print the missing character on output. If any values are negative, the minus sign is counted as a place before the decimal.

Note that in Appendix C, on the PRINT output, the last listing is the MINIMAL dataset which was copied from the SAMPLE INSSD. It is apparent from the VB1-VB3 variables that using BESTw. does not guarantee that the decimal points will line up in columns, nor that all the data will be displayed with the same precision (including significant trailing zeros). VRANGE solves these problems, with the help of BESTw. The formats created by VRANGE are shown in Appendix C as the file LIST, and also in the LOG in Appendix B, where MPRINT shows the FORMAT statement in DATA MINIMAL generated by VRANGE shortly after line 133. (Some of the LOG was deleted to save space.)

(Note also that this same procedure can be written without macros, by a DATA _NULL_ and FILE TMP where TMP is a FILENAME that points to a file that is INCLUDE'd at a subsequent point. This has both advantages and disadvantages. The advantages are that (1) the LOG will have line numbers and indentation (as specified by the PUT statements), which is very useful

for debugging, using error messages, and documentation. On the other hand, it is harder to modify the code, since it is enclosed in quotes, and thus sometimes a line must be constructed using a temporary character variable (CTMP) followed by a PUT CTMP.)

Some improvements needed include (1) determining whether any of the named VARS are names of variables used in the DATA &OUTSSD step created by VRANGE, (2) checking that BEST32 does not create an exponential form (e.g., 1E-300), and (3) checking that the minimal format width is not greater than 32. The first problem could be solved by adding another macro parameter named SUFFS that would provide suffixes which VRANGE would then use to create distinct names. The second problem could be solved by using a "variable format", where one checks for this problem (for each value of a variable) with BESTw to see if an E-format is generated, in which case one might use an Ew.d format rather than the minimal format from VRANGE. (But this would prevent the use of PROCs to print data, because a variable format is needed for the special cases. It would be nice to have an "indirect" format capability, where the format specified would be a character variable rather than a constant.) The third problem might be resolved in a manner similar to the second, using a variable format.

In summary, VRANGE produces a useful minimal format based on BEST32. Some improvements would extend its usefulness.

References

Barlow, Ginger, and Ringelberg, Angela S. "Dances with Data." *Proceedings of the Seventeenth Annual SAS® Users Group International Conference*, Cary, NC: SAS Institute Inc., 1992.

Goldstein, Richard, Section Editor. "Editor's Notes." *The American Statistician*, 46-4, November 1992, 319-320.

SAS Institute Inc. *SAS® Language Guide for Personal Computers, Release 6.03 Edition*. Cary, NC: SAS Institute Inc., 1988.

Acknowledgments

SAS is a registered trademark of SAS Institute Inc.

MS-DOS is a registered trademark of Microsoft Corporation.

Address Correspondence to:

Andrew E. Barnett
Clinical Trials Support Services
P.O. Box 1815
Burlington, NC 27216-1815
(919) 222-8030

Appendix A -- VRANGE.SAS source code

```
*-----;
*  FILENAME:  VRANGE.SAS                                *;
*  VERSION:   V4                                          *;
*  PURPOSE:   VRANGE determines minimal formats for variables from *;
*             a given dataset. A minimal format is data- *;
*             dependent, and is the minimum field needed to print all *;
*             data in decimal format. The decimal points on the output *;
*             will line up, unlike using BESTw.. This method uses BESTw. to *;
*             convert data from internal representation to decimal, and uses *;
*             the decimal representation to determine minimum field width. *;
*  AUTHOR:    A.E.Barnett                                *;
*  CREATION DATE: 03-SEP-1992                             *;
*  MODIFICATIONS:                                         *;
*  NOTES/ASSUMPTIONS:                                     *;
*      -- VRANGE will not determine the actual precision of the values *;
*         (as opposed to the precision of the variables) if all of *;
*         the values have zero in the final decimal place(s). *;
*  INPUT:                                             *;
*      -- INSSD=name of input data set                    *;
*      -- OUTSSD=name of output dataset to contain formats *;
```

```

*   -- VARS=list of names of numeric variables to be processed      *;
*   -- MINIMAL=dataset name for output copy of OUTSSD with minimal *;
*   formats (leave blank if no dataset wanted)                      *;
* OUTPUT:                                                            *;
*   -- dataset OUTSSD contains minimal formats for each input      *;
*   variable in VARS, with the same names as in VARS              *;
* SAMPLE CALL: %VRANGE(INSSD,OUTSSD,LONGNAME,MINIMAL)               *;
*   -- this will produce one variable in OUTSSD named LONGNAME, and *;
*   a new copy of INSSD with all variables and with LONGNAME      *;
*   FORMATTed to its minimal format.                                *;
* INCLUDES: --none--                                                *;
* LIBNAMES: --none--                                                *;
* CHANGES NEEDED:                                                  *;
*   -- Have VRANGE check that VARS names are not used by VRANGE   *;
*   -- Solve problems of E-format output from BEST32., and possible *;
*   width greater than 32 (w.d, where w=IVARS+FVAR+1).            *;
*-----*
      %macro vrange(inssd,outssd,vars,minimal);
%*Count variables listed in vars;
      %let n=0; %let name=%scan(&vars,&n+1);
      %do %while ("%name" ne "");
          %let n=%eval(&n+1); %let name=%scan(&vars,&n+1);
      %end;
%*Generate minimal formats;
      data &outssd (keep=fmt1-fmt&n
          rename=(
%*Rename formats to same names as original numeric variables;
          %do i=1 %to &n; %let var=%scan(&vars,&i); fmt&i=&var %end;));
%*Read original data values;
      set &inssd (keep=&vars) end=eof;
%*Store values in VARS;
      array vars{*} &vars;
%*IVARS is maximum number of decimal places (including sign) before decimal;
      array ivars{*} ivars1-ivars&n;
%*FVAR is maximum number of decimal places (including exponent) after decimal;
      array fvars{*} fvars1-fvars&n;
%*CVARS is BEST32. representation of VARS;
      array cvars{&n} $32 temporary;
%*FMT is minimal format of form w.d, where w=IVARS+FVAR+1 and d=FVAR;
      array fmt{*} $8 fmt1-fmt&n;
%*Retain values of IVARS and FVAR;
      retain
      %do i=1 %to &n %by 5; %do j=(&i-1)*5+1 %to &i*5;
          %if &j le &n %then ivars&j; %end; %end; ;
      retain
      %do i=1 %to &n %by 5; %do j=(&i-1)*5+1 %to &i*5;
          %if &j le &n %then fvars&j; %end; %end; ;
      if n=1 then do i=1 to &n; ivars{i}=0; fvars{i}=0; end;
      do i=1 to &n;
          if vars{i} gt .z then do;
              cvars{i}=left(put(vars{i},best32.));
              j=index(cvars{i},'.' );
              ivars{i}=max(j-1,ivars{i});
              fvars{i}=max(length(cvars{i})-j,fvars{i});
          end;
      end;
%*On end of file, create minimal formats for each variable, and output;
      if eof then do;
          do i=1 to &n;
              fmt{i}=compress(ivars{i}+fvars{i}+1)!!"."!!
                  compress(fvars{i});
          end;

```

```

        output;
    end;
run;
*Print minimal formats;
proc print data=&outssd;
    title2 "OUTSSD=&outssd -- from INSSD=&inssd";
run;
title2;
    %if "&minimal" ne "" %then %do;
*Store formats in macro variables;
    data _null_;
        array fmts{*} $8 &vars;
        set &outssd;
        do i=1 to &n; call symput('fmts'!!compress(i),fmts{i}); end;
        stop;
    run;
*Create copy of &OUTSSD with new formats;
    data &minimal;
        %do i=1 %to &n %by 5;
            format
                %do j=(&i-1)*5+1 %to &i*5;
                    %if &j le &n %then %do;
                        %let var=%scan(&vars,&j); &var &&fmts&j;
                    %end;
                %end;;
            %end;
        set &inssd;
    run;
    %end;
%mend vrange;

```

Appendix B -- Sample LOG output

```

1      options ls=80 ps=60 pageno=1 nodate center;
2      options mprint nosymbolgen nomtrace source2;
3      %let pid=VRANGE4;
4      %let src=C:\SESUG93\;
5      %let nob=3; %let nvars=3; %let w=4; %let w2=%eval(&w*2+1);
6      title "&src.&pid..SAS -- SAS&sysver";
7      filename tmp 'tmp.sas';
105     data sample (keep=v1-v&nvars);
106         array v{*} v1-v&nvars;
107         format v1-v&nvars best16.;
108         do iobs=1 to &nobs;
109             retain nseed 1 useed 2;
110             do iv=1 to &nvars;
111                 n=normal(nseed); u=uniform(useed);
112                 if iv=1 then do; if u gt .5 then v{iv}=.z; else v{iv}=.; end;
113                 else if iv=2 and u gt .5 then v{iv}=.;
114                 else do;
115                     v{iv}=round(n*1E&w,10**(-int(u*&w+1.5)));
116                 end;
117             end;
118             output;
119         end;
120         file tmp;
121         put '      %vrange(sample,list,'
122             /'          '@;
123         do i=1 to &nvars; put 'v' i @; end; put ',';
124         put '      minimal);';
125         stop;

```

```

126      run;
127      proc print data=sample;
128          title2 'Sample Data';
129      run;
130      %include tmp / source source2;
131+      %vrange(sample,list,
132+          v1 v2 v3 ,
133+          minimal);
MPRINT(VRANGE):  DATA MINIMAL;
MPRINT(VRANGE):  FORMAT V1 1.0 V2 6.1 V3 11.5 ;
MPRINT(VRANGE):  SET SAMPLE;
MPRINT(VRANGE):  RUN;
134      data minimal;
135          set minimal;
136          array v{*} v1-v&nvars;
137          array vb{*} vb1-vb&nvars;
138          do i=1 to &nvars; vb{i}=v{i}; end;
139      run;
140      proc print data=minimal;
141          title2 "Minimal compared to BEST&w2..";
142          format vb1-vb3 best&w2..;
143          var v1 vb1 v2 vb2 v3 vb3;
144      run;

```

Appendix C -- PRINT output

C:\SESUG93\VRANGE4.SAS -- SAS6.04 1
Sample Data

OBS	V1	V2	V3
1	.	.	3965.7686
2	Z	.	-6242.32294
3	.	-866.1	-5941.8

C:\SESUG93\VRANGE4.SAS -- SAS6.04 2
OUTSSD=list -- from INSSD=sample

OBS	V1	V2	V3
1	1.0	6.1	11.5

C:\SESUG93\VRANGE4.SAS -- SAS6.04 3
Minimal compared to BEST9.

OBS	V1	VB1	V2	VB2	V3	VB3
1	.	.	.	.	3965.76860	3965.7686
2	Z	Z	.	.	-6242.32294	-6242.323
3	.	.	-866.1	-866.1	-5941.80000	-5941.8