

Using Cardinality Ratio for Fast Data Review

Ronald J. Fehd, Stakana Analytics, Atlanta, GA, USA

Abstract

Description : This paper reviews the database concept Cardinality Ratio. The SAS® frequency procedure can produce an output data set with a list of the values of a variable. The number of observations of that data set is called N-Levels. The quotient of N-Levels divided by the number-of-observations of the data is the variable's Cardinality Ratio (CR). Its range is in (0–1] and four categories of variables' CR are obvious: useless, discrete, continuous and unique.

Purpose : Cardinality Ratio provides an important value during data review. The four groups of values are used to create control data sets which can automate data review. Macros are provided that facilitate fast data review.

Audience : data managers and programmers.

Programs : in this paper are available in Fehd [5, sco.Cardinality-Ratio]

Keywords : continuous, database, dimensionless, discrete, fact, frequency, keys: foreign or primary, nlevels, number of observations (nobs), unique

Quote : Information is *the* difference
that makes *a* difference. Gregory Bateson 1904–1980
Steps to an Ecology of Mind, 1972
italics added by R.J. Fehd

Contents

Programs	4
cr-calc.sas	4
listvars.sas	7
cr-data.sas	8
cr-freq.sas	9
cr-mode.sas	10
cr-smry.sas	10
Unit Tests	12
cr-calc-test-class.sas	12
cr-calc-test-class-with-format.sas	13
cr-calc-test-shoes.sas	13
cr-calc-test-sashelp-all.sas	14
Summary	15
Bibliography	16

Introduction

Overview

When starting a new project a programmer or data detective may use several procedures during data discovery to confirm the relationships between variables in a data set.

The cardinality ratio (CR) of a variable is the quotient of the number of levels of that variable divided by the number of rows of the data set. The dimension: n-rows of numerator and denominator, cancels out leaving a pure number in the range of >zero – one.

CR is similar in concept to the log function: it reduces large numbers to a finite range which makes comparisons easier to grasp.

The Sets of Values

CR can be grouped into four categories.

- continuous : $CR \gtrsim 0.5$
- discrete: $CR \lesssim 0.5$
- unique: $CR = 1$
- single-valued: $n\text{-levels} = 1$

Note: One-half (0.5) is an arbitrary separation value.

continuous : information: is.a fact variable; if numeric can be summarized

discrete : indicators: character variables have standardized case: either upper or lower; numerics may be integers, or in a small finite range

information: is.a classification variable;

todo: locate one-to-one formats, or dimension (lookup) tables

unique : variable is a row-identifier; if numeric and the range is exactly 1:n-observations then it is the row-number

information: is.a primary key;

single-valued : values may be:

- character: blank
- numeric: missing
- a single value, indicating data set is a subset of larger data

information: worthless, discard

Task

The task of this article is to show that the mean of the cardinality ratios separates the discrete from the continuous; the former is summarized with the frequency procedure and the latter with the summary procedure.

Continued on next page.

Algorithm of Calculator

These are the steps in the cardinality-ratio calculator program.

1. `listvars`: make list of variable names from `contents`
2. `freq`: make list of n-levels
out= print n-levels
3. sort and merge `contents` and frequency output data sets
calculate cardinality ratio
out=CR-list
4. calculate mean of CR
5. add mean to CR-list
assign categories: unique, discrete, continuous
make lists for frequency of discrete,
mode of either, and summary of continuous
6. print CR-list with categories
7. for each list, call macros of frequency, mode, summary,
out = CR-summaries
8. print CR-summaries

Note: This algorithm is from Fehd [7], `SmryEachVar` and Fehd [9], `Cardinality Ratio`.

Algorithm of Procedures

These are the steps in the cardinality-ratio summarization routines, `cr-frequency`, `cr-mode` and `cr-summary`.

1. procedure, out=out-from-proc
2. data for-append, standardize data structure: `cr_data`
3. append
base = CR-summaries
data = out-from-proc

Common Data Structure

This is the common data structure provided by the subroutine `cr-data`.

```

1 create table WORK.OUT_FREQUENCIES( bufsize=16384 )
2   libname char(8),
3   memname char(32),
4   name char(32),
5   valu_c char(32),
6   valu_txt char(32),
7   valu_n num,
8   count num format=COMMA. label='N',
9   percent num format=6.2 label='-%-'
```

Notes: Each procedure renames the variable according to its type. Character variables are renamed to `valu_c`; numeric variables are renamed to `valu_n`; the mode procedure and formatted values are written into `valu_txt`.

Programs

Overview

This section lists the suite of programs that produce a report of the cardinality ratios of variables in a data set.

- cardinality ratio calculator
- listvars, make list of variables from contents
- cr-data, standardized data structure for cr-*
- cr-frequency
- cr-mode, proc mode is `proc freq order = freq`
- cr-summary

Note: The listing of list-processing macro `callmacr.sas` is in Fehd [11].

cr-calc.sas

```

1  /*          name: <UNC>\SAS-site\macros\cr_calc.sas
2  author: Ronald J. Fehd 2012, 2013
3  -----
4  Summary      : description  : cardinality ratio calculator
5                purpose      : data review
6  -----
7  Contexts     : program group: list processing program
8                program type: module
9                SAS         type: macro routine
10               uses routines : callmacr to call:
11               uses subroutines: cr_freq, cr_mode, cr_smry
12  -----
13  Specifications: input   : required: libname, memname
14                process: contents out=
15                        freq nlevels out=
16                data: merge contents + freq,
17                        calculate CR
18                summary: mean of CR
19                data: add category is_a,
20                        output various
21                output: save and print: out-lib.out-data
22  -----
23  Parameters   : libname  : actually the libref
24                memname  : data set name
25                out_lib  : work | library
26                out_data : check the default: cr_&memname
27                out_smry : &out_lib..smry_&out_data
28  -----
29  Usage Example:
30  *options mprint source2;
31  %cr_calc(libname = sashelp
32           ,memname = shoes)
33  Notes:
34  * certain data sets will generate these warnings:
35  WARNING: The variable nmisslevels in the DROP, KEEP,
36           or RENAME list has never been referenced.
37  WARNING: The variable nnonmisslevels in the DROP, KEEP,
38           or RENAME list has never been referenced.
39  when present, these variables are in:
40  create table FREQ_N_LEVELS
41             (label='Number of Variable Levels')
42             name char(14) label='Table Variable',
43             TableVarLabel char(28) label='Table Variable Label',
44             n_levels num format=BEST8. label='Number of Levels',
45             NMissLevels num format=BEST8.
46                 label='Number of Missing Levels',
47             NNonMissLevels num format=BEST8.

```

```

48          label='Number of Nonmissing Levels'
49
50 * this is a Derivative Work of these suites:
51 ** SmryEachVar
52 http://www2.sas.com/proceedings/forum2008/003-2008.pdf
53 http://www.sascommunity.org/wiki/
54          SmryEachVar_A_Data_Review_Suite
55 ** Cardinality Ratio
56 http://www.sascommunity.org/wiki/Cardinality_Ratio
57 DateTime 8/7/2014 10:21:39 AM
58 Words          : 852
59 Lines          : 256
60 Characters (no spaces) : 5416
61 Characters (with spaces): 8481
62 *****/
63 %MACRO cr_calc
64     (libname      = sashelp
65      ,memname     = class
66      ,out_lib     = work
67      ,out_data    = cr_&memname
68      ,out_smry    = cr_&memname._smry
69      ,mode_show   = 3
70      ,mode_nlevels = 33 /* fib: 1 1 2 3 4 8 13 21 33 */
71      ,testing     = 0)
72 / des = 'site: cardinality ratio calculator';
73 %local out_contents out_freq out_from_procs;
74 %let out_contents = list_names;
75 %let out_freq     = freq_n_levels;
76 %let out_from_procs = cr_summaries;
77 %let testing = %eval( not(0 eq &testing)
78                   or %sysfunc(getoption(mprint)) eq %upcase(mprint));
79
80 %listvars(data      = &libname..&memname
81           ,drop_keep =
82             keep = libname memname name type format nob
83           ,out      = &out_contents)
84
85 **** make data set with nlevels;
86 PROC freq data = &libname..&memname
87     nlevels ;
88     ods      exclude onewayfreqs;
89     ods      output
90     nlevels = &out_freq
91     (rename = (tablevar = name
92              nlevels = n_levels));
93     title3 "n-levels of &libname..&memname";
94
95 %if &testing %then %do;
96     PROC sql; describe table &syslast.;
97         quit;
98     %end;
99
100 **** prepare for merge;
101 PROC sort data = &out_contents
102     out = &out_contents;
103     by  libname memname name;
104
105 PROC sort data = &out_freq
106     out = &out_freq;
107     by  name;
108
109 DATA &out_data
110     (keep = card_ratio libname memname name type
111      format label n_levels nob
112      nmisslevels nnonmisslevels);
113     if 0 then do;
114         attrib card_ratio length = 8
115             label = 'cardinality ratio';
116         set &out_contents (obs = 0);
117         set &out_freq     (obs = 0);
118         end;
119
120 do until(endofile);
121     merge &out_contents
122         &out_freq      end = endofile;
123     by  name;
124     card_ratio = n_levels/nobs;

```

```

125     libname     = lowercase(libname);
126     memname    = lowercase(memname);
127     output;
128     end;
129 call symputx('nobs',nobs);
130 stop;
131 run;
132
133 %if &testing %then %do;
134     PROC sql; describe table &syslast.;
135         quit;
136     %end;
137
138 **** calculate middle of CR range:
139     separates foreign keys::discrete and facts::continuous;
140 PROC summary data = &out_data
141     mean;
142     var    card_ratio;
143     output
144     out = mean
145     (keep = mean)
146     mean (card_ratio) = mean;
147
148 %if &testing %then %do;
149     PROC sql; describe table &syslast;
150     quit;
151 %end;
152
153 **** save mean for use in calculation, label and title;
154 PROC sql noprint;
155     select mean
156     into :mean
157     from &syslast;
158     quit;
159 %put note: &sysmacroname mean: &mean;
160 run;
161 %if &mean eq 1 %then %do;
162     %put note: &sysmacroname exit: mean eq 1;
163     %return;
164 %end;
165
166 %let drop_list = drop = is_a card_ratio;
167 DATA &out_data
168     cr_list_freq(&drop_list)
169     cr_list_mode(&drop_list)
170     cr_list_smry(&drop_list);
171     keep is_a card_ratio libname memname name type format;
172     if 0 then do;
173         attrib is_a length = $ %length(_primary key?!);
174         set &out_data
175             (keep = card_ratio n_levels
176              libname memname name type format);
177     end;
178     set mean;
179
180 do until(endofile);
181     set &out_data end = endofile;
182     if card_ratio eq 1 then
183         is_a = 'primary key?!';
184     else if n_levels eq 1 then
185         is_a = 'nlevels=1';
186     else if card_ratio le mean then do;
187         is_a = 'foreign key';
188         if n_levels le &mode_nlevels then output cr_list_freq;
189         else output cr_list_mode;
190     end;
191     else if card_ratio gt mean then do;
192         is_a = 'fact';
193         if type = 'c' then output cr_list_mode;
194         else if type = 'n' then output cr_list_smry;
195     end;
196     output &out_data;
197 end;
198 stop;
199 run;
200
201 %if &testing %then %do;

```

```

202 PROC sql; describe table &syslast.;
203 quit;
204 *proc print data =cr_list_freq;title3 cr_list_freq;
205 *proc print data =cr_list_mode;title3 cr_list_mode;
206 *proc print data =cr_list_smry;title3 cr_list_smry;
207 %end;
208
209 PROC sort data = &out_data
210 out = &out_lib..&out_data
211 (label=
212 "&out_lib..&out_data nobs=&nobs mean=&mean");
213 by is_a name card_ratio;
214
215 PROC print data = &out_lib..&out_data label;
216 title3 "cardinality ratios of &libname..&memname"
217 " nobs=&nobs mean=&mean";
218 by is_a;
219 id is_a;
220 run;
221 **** use routine to call subroutines;
222 %callmacr(data = cr_list_freq
223 ,macro_name = cr_freq
224 ,hex16 = 0)
225 *** note: hex16 = 0 :: pass varnum as integer;
226 %callmacr(data = cr_list_mode
227 ,macro_name = cr_mode
228 ,macro_parms = %nrstr(mode_hi_lo=&mode_show)
229 ,hex16 = 0
230 )
231 %callmacr(data = cr_list_smry
232 ,macro_name = cr_smry
233 ,hex16 = 0
234 )
235 *callmacr(data = cr_list_smry
236 ,macro_name = cr_univ
237 ,hex16 = 0
238 );
239 %if not(%sysfunc(exist(&out_from_procs))) %then %do;
240 %put note: 0.1 &sysmacroname exit not exist(
241 &out_from_procs);
242 %return;
243 %end;
244 PROC datasets library = work
245 nodetails nolist;
246 change
247 &out_from_procs = &out_smry;
248 %if %lowercase(&out_lib) ne work %then %do;
249 PROC copy in = work
250 out = &out_lib;
251 select &out_smry;
252 %end;
253 proc print data = &out_lib..&out_smry label;
254 title4 &out_lib..&out_smry;
255 run;
256 %mend cr_calc;

```

listvars.sas

```

----- listvars.sas -----
1 /* name: <UNC>\SAS-site\macros\listvars.sas
2 Make List of Variable Names
3 description: make list of variable names
4 purpose: prepare list processing data set
5 input : data : _all_ | data set name
6 drop_keep: keep=libname memname name type
7 where : 1 | logical expression
8 process: proc contents
9 output: out =
10 usage:
11 %listvars(data = sashelp.class
12 ,drop_keep =
13 keep = libname memname name type format)
14 see also: http://www.sascommunity.org/wiki/Making\_Lists **/
15 %MACRO listvars
16 (data = sashelp.class
17 ,drop_keep = drop=type_n formatd formatl

```

```

18     ,where      = 1
19     ,out       = list_vars
20     ,testing   = 0)
21 / des = 'site from contents make list of variable names';
22 %let testing = %eval( not(0 eq &testing)
23     or %sysfunc(getoption(mprint)) eq %upcase(mprint));
24
25 PROC contents data = &data
26     noprint
27     out = &out
28     (keep = libname memname memtype varnum
29     name type length label format
30     formatd formatl nob)
31     rename = (type = type_n)
32     where = (&where
33     and memtype eq 'DATA' );
34
35 DATA &out (&drop_keep
36     label = "list vars of &data");
37     attrib libname    length = $ 8 label = 'libname'
38     memname    length = $ 32 label = 'mem name'
39     varnum     length = 8 label = 'var num'
40     name_lc    length = $ 32 label = 'low (var name)'
41     name       length = $ 32 label = 'var name'
42     type       length = $ 1 label = 'type'
43     length     length = 8 label = 'length'
44     format     length = $ 34 label = 'format'
45     label      length = $256 label = 'label'
46     nob       length = 8 label = 'n obs';
47
48 do until(endofile);
49     set &out(drop = memtype)
50     end = endofile;
51     libname = lowercase(libname);
52     memname = lowercase(memname);
53     name_lc = lowercase( name);
54     * align format, type to sql.dictionary.columns:
55     1. convert type from numeric to $charl
56     2. add char.prefix $ and suffix dot to formats;
57     if format ne ' ' then do;
58         if formatl then format = cats(format,formatl,'.');
59         else format = cats(format
60             ,'.');
61         if formatd then format = cats(format,formatd);
62     end;
63     /*****
64     if format ne ' ' then
65         format = catt(format,'.');
66     if format in ('$', '$.') then
67         format = ' ';*****/
68     if type_n eq 1 then type = 'n';
69     else if type_n eq 2 then type = 'c';
70     output &out;
71     end;
72 stop;
73 run;
74 %if &testing %then %do;
75     proc sql; describe table &out;
76     quit;
77     %end;
78 %mend listvars;

```

cr-data.sas

```

1  /*          cr_data.sas
2  /*          name: <UNC>\SAS-site\macros\cr_data.sas
3  /*          author: Ronald J. Fehd 2013
4  Summary : description : common data structure of cr_*
5           purpose      : standardization
6 Contexts: program group: list processing program
7           program type: subroutine
8           SAS type: macro subroutine
9           called by routines cr_* *****/
10 %MACRO cr_data(_)
11 / des = 'site: card. ratio suite: common data structure';
12     keep  _libname _memname _name
13         _valu_c _valu_txt _valu_n _count _percent;

```



```

13     attrib _libname length = $8 label = 'libname'
14     _memname length = $32 label = 'memname'
15     _name length = $32 label = 'name'
16     _valu_c length = $32 label = 'valu c'
17     _valu_txt length = $32 label = 'valu txt'
18     _valu_n length = 8 label = 'valu n'
19     _count length = 8 format = comma. label = 'N'
20     _percent length = 8 format = 6.2 label = '-%-' ;
21     retain _libname "%lowercase(&libname)"
22     _memname "%lowercase(&memname)"
23     _name "&name"
24     _valu_c '.' _valu_txt '..'
25     _valu_n . _count . _percent . ;
26 %mend cr_data;

```

cr-freq.sas

```

1          cr_freq.sas
2  /* name: cr_freq.sas
3     author: Ronald J. Fehd 2013
4  Summary      : description : proc freq
5                purpose   : standardize output
6  Contexts    : program group: list processing program
7                program type: routine
8                SAS       type: macro routine
9  Specifications: input   : libname memname name type format
10                process: freq out= ***/
11 %macro cr_freq
12     (libname      =
13      ,memname     =
14      ,name        =
15      ,type        =
16      ,format      =
17      ,out_append = cr_summaries
18      ,out_data    = out_frequencies
19      ,testing     = 0)
20 / des = 'site: list processing proc freq';
21 %let testing = %eval( not(0 eq &testing)
22                or %sysfunc(getoption(mprint)) eq %upcase(mprint));
23 PROC freq data = &libname.&memname;
24     format &name;*remove formats;
25     tables &name
26           / list missing noprint
27           out = &out_data
28           (rename = (&name = _valu_&type
29                     count = _count
30                     percent = _percent));
31 DATA &out_data;
32     %cr_data(attributes);
33
34 do until(endofile);
35     set &out_data
36       end = endofile;
37     %if &format ne %then %do;
38         _valu_txt = put(_valu_&type,&format);
39     %end;
40     output;
41 end;
42 stop;
43 run;
44
45 %if &testing %then %do;
46     PROC sql; describe table &syslast;
47     quit;
48     %end;
49
50 PROC append base = &out_append
51             force
52             data = &out_data;
53 PROC delete data = &out_data;
54 run;
55 %mend cr_freq;

```

cr-mode.sas

```

1  /*      name: <UNC>\SAS-site\macros\cr_mode.sas
2  author: Ronald J. Fehd 2013
3  Summary   : description   : calculate modes, hi and low
4             : purpose      : standardize output
5  Contexts  : program group: list processing program
6             : program type: routine
7             : SAS         : type: macro routine
8  Specifications: input   : libname, memname, name, type
9             : process:   : freq order=freq out=   ***/
10 %macro cr_mode
11     (libname      =
12      ,memname     =
13      ,name        =
14      ,type        =
15      ,format      =
16      ,mode_hi_lo = 3
17      ,out_append  = cr_summaries
18      ,out_data    = out_modes
19      ,testing     = 0)
20 / des = 'site: card. ratio suite: proc mode';
21 %let testing = %eval( not(0 eq &testing)
22                  or %sysfunc(getoption(mprint)) eq %upcase(mprint));
23
24 PROC freq data = &libname..&memname
25             order = freq;
26             format &name;*remove formats;
27             tables &name
28                 / list missing noprint
29             out = &out_data
30             (rename = (&name = _valu_&type
31                      count = _count
32                      percent = _percent));
33
34 DATA &out_data;
35     %cr_data(attributes);
36
37 do rownmbr = 1 to &mode_hi_lo;
38     set &out_data nobs = n_rows point = rownmbr;
39     _valu_txt = "mode: high &mode_hi_lo";
40     output;
41     end;
42 do rownmbr = n_rows +1 - &mode_hi_lo to n_rows;
43     set &out_data nobs = n_rows point = rownmbr;
44     _valu_txt = "mode: low &mode_hi_lo";
45     output;
46     end;
47 stop;
48 run;
49
50 %if &testing %then %do;
51     PROC sql; describe table &syslast;
52             quit;
53     %end;
54
55 PROC append base = &out_append
56             force
57             data = &out_data;
58 PROC delete data = &out_data;
59 run;
60 %mend cr_mode;

```

cr-smry.sas

```

1  /*      name: cr_smry.sas
2  author: Ronald J. Fehd 2013
3  Summary   : description   : summarize numeric variable
4             : purpose      : standardize output
5  Contexts  : program group: list processing program
6             : program type: routine
7             : SAS         : type: macro routine
8  Specifications: input   : libname memname name type format
9             : process:   : summary out=   ***/

```

```

10 %macro cr_smry
11     (libname     =
12     ,memname    =
13     ,name       =
14     ,type       =
15     ,format     =
16     ,out_append = cr_summaries
17     ,out_data   = out_summaries
18     ,testing    = 0)
19 / des = 'site: card. ratio suite: proc summary';
20 %let testing = %eval( not(0 eq &testing)
21     or %sysfunc(getoption(mprint)) eq %upcase(mprint));
22
23 PROC summary data = &libname..&memname;
24     format &name;*remove formats;
25     var &name;
26     output
27         out = summary
28         ( drop = _type_ _freq_)
29     n (&name) = n
30     nmiss (&name) = nmiss
31     min (&name) = min
32     std (&name) = std
33     p25 (&name) = p25 %*q1;
34     mean (&name) = mean %*average;
35     median (&name) = median %*p50::middle;
36     p75 (&name) = p75 %*q3;
37     max (&name) = max ;
38
39 PROC transpose data = summary
40     out = summary_t
41     (keep = coll _valu_c
42     rename =(coll = _valu_n ))
43     name = _valu_c ;
44 DATA &out_data;
45     %cr_data(attributes);
46     if 0 then set &libname..&memname(obs=0)
47         nobs = nobs_data;
48
49 do until(endofile);
50     set summary_t end = endofile;
51     if _valu_c in ('n','nmiss') then
52         _percent = 100*( _valu_n/nobs_data);
53     else _percent = .;
54     %if &format ne %then %do;
55         _valu_txt = put(_valu_&type,&format);
56     %end;
57     output;
58 end;
59 stop;
60 run;
61
62 %if &testing %then %do;
63     PROC sql; describe table &syslast;
64         quit;
65     %end;
66
67 PROC append base = &out_append
68     force
69     data = &out_data;
70 PROC delete data = &out_data;
71 PROC delete data = summary;
72 PROC delete data = summary_t;
73 run;
74 %mend cr_smry;

```

Unit Tests

Overview

This section lists the set of programs used to do unit tests of the cardinality-ratio-calculator suite.

- cr-calc-test-class
 - cr-calc-test-class-with-format, shows `valu.txt` with format
 - cr-calc-test-shoes, shows mode in `valu.txt`
 - cr-calc-test-sashelp-all, review all data sets in libref
-

cr-calc-test-class.sas

program

```

1 options mprint source2;
2 %cr_calc(libname = sashelp
3         , memname = class)

```

report

```

1 |Number of Variable Levels
2 |Variable  Levels
3 |-----  -
4 |Name      19
5 |Sex       2
6 |Age       6
7 |Height    17
8 |Weight    15
9 |
10 |cardinality ratios of sashelp.class nobs=19 mean=0.621053
11 |      card_
12 |is_a    ratio  libname memname name  type  format
13 |primary key?! 1.00000  sashelp  class  Name   c
14 |
15 |fact      0.89474  sashelp  class  Height n
16 |          0.78947  sashelp  class  Weight  n
17 |
18 |foreign key 0.31579  sashelp  class  Age     n
19 |          0.10526  sashelp  class  Sex     c
20 |
21 |work.cr_class_smry
22 |libname memname name  valu_c valu_txt  valu_n count percent
23 |
24 |sashelp class Age     .      .      11.000  2    10.53
25 |sashelp class Age     .      .      12.000  5    26.32
26 |...
27 |sashelp class Age     .      .      16.000  1    5.26
28 |sashelp class Sex     F      .      .      9      47.37
29 |sashelp class Sex     M      .      .      10     52.63
30 |sashelp class Height  n      .      .      19.000 .    100.00
31 |sashelp class Height  nmiss .      .      0.000 .    0.00
32 |sashelp class Height  min   .      .      51.300 .    .
33 |sashelp class Height  mean  .      .      62.337 .    .
34 |sashelp class Height  median .      .      62.800 .    .
35 |...
36 |sashelp class Height  max   .      .      72.000 .    .
37 |sashelp class Weight  n      .      .      19.000 .    100.00
38 |sashelp class Weight  nmiss .      .      0.000 .    0.00
39 |...
40 |sashelp class Weight  max   .      .      150.000 .    .

```

Continued on next page.

cr-calc-test-class-with-format.sas

program This test program shows a report of a variable with a format.

```

1 options mprint source2;
2 proc format;
3 value $sex 'F' = 'female'
4           'M' = 'male'
5           %* 'I' = 'intersex' %*;
6           %* other = 'unknown' %*;
7           ;
8 data class_with_format;
9     if 0 then set sashelp.class;
10    attrib sex length = $1 format = $sex.;
11 do until(EndOfFile);
12    set sashelp.class end = endofile;
13    output;
14  end;
15 Name = 'Chris'; Sex = 'I';
16 output;
17 stop;
18 run;
19 proc sql; describe table &syslast;
20     quit;
21 %cr_calc(libname = work
22         ,memname = class_with_format)

```

report Formats of values are in variable valu_txt.

```

1 | cardinality ratios of work.class_with_format nobs=20 mean= 0.61
2 |         card_
3 | is_a      ratio libname      memname      name type format
4 | ...
5 | foreign key 0.15 work      class_with_format sex c $SEX.
6 |
7 | libname memname      name valu_c valu_txt valu_n count percent
8 | work      class_with_format sex      F female . 9 45.00
9 | work      class_with_format sex      I I . 1 5.00
10 | work      class_with_format sex      M male . 10 50.00

```

cr-calc-test-shoes.sas

program This test program shows output from the mode procedure.

```

1 options mprint source2;
2 %cr_calc(libname = sashelp
3         ,memname = shoes)

```

report Variable Subsidiary is a foreign key with n-levels greater than 33, the default value of cr_calc parameter mode_nlevels; instead of proc freq, the mode macro is used. Note the high values in variable valu_txt.

```

1 | Variable      Label              Levels
2 | ...
3 | Subsidiary                    53
4 |         card_
5 | is_a      ratio libname memname name      type      format
6 | ...
7 | foreign key 0.13418 sashelp shoes Subsidiary c
8 |
9 | libname memname name      valu_c      valu_txt      valu_n count percent
10 | sashelp shoes Subsidiary Addis Ababa mode: high 3 . 8 2.03
11 | sashelp shoes Subsidiary Al-Khobar mode: high 3 . 8 2.03
12 | sashelp shoes Subsidiary Budapest mode: high 3 . 8 2.03

```

Continued on next page.

cr-calc-test-sashelp-all.sas

program This program is a stress test. It runs `cr_calc` on each data set in *libref* *sashelp*. This programs duplicates the *SmryEachVar* suite, *Fehd* [7].

```
cr_calc-test-sashelp-all.sas
1 * name: <UNC>\SAS-site\sas-macros\cr_calc-test-sashelp-all.sas
2 description: stress test of cr_calc
3   purpose: duplicate SmryEachVar:
4     summarize all data sets in libref ***/
5 options mprint source2;
6
7 %let in_libref = sashelp;
8
9 *** make list of variables;
10 %listvars(data      = &in_libref._all_
11           ,drop_keep =
12             keep = libname memname name type format nobs
13           ,where     = nobs
14           ,out       = list_vars_from_contents)
15
16 *** make list of memnames from list_vars;
17 proc sort data = &syslast
18         (keep = libname memname)
19         out  = list_memnames
20         nodupkey;
21         by   libname memname;
22
23 proc print data = &syslast;
24         title3 "&syslast of &in_libref";
25 run;
26 %callmacr(data      = &syslast
27           ,macro_name = cr_calc)
```

Summary

Conclusion

Cardinality Ratio is valuable information to have in data review. Its small finite range is easier to parse for meaning than the constantly changing and larger number-of-observations (nobs) of the data set. The mean of CR offers an easy way to differentiate between discrete and continuous variables, which are processed with frequency for the former and summary for the latter.

Further Reading

- Programs : for this paper are in Fehd [5, sco.Cardinality-Ratio]
- Predecessors : This paper was first published in Fehd [9, sfg2013.299].
- Fehd [7, sfg2008.003] (SmryEachVar) developed a suite of programs to return a list of the frequencies of each variable in a data set or libref.
- Cardinality Ratio is identified in Fehd [8, wuss2008.Database-Vocabulary] for which SmryEachVar is the predecessor.
- Programs for SmryEachVar which includes calculations for Cardinality Ratio are here: Fehd [6, sco.SmryEachVar].
- Call-Macro : The list processing routine Call-Macro is described in Fehd [11]
- Theory : The macros in this article illustrate the macro design ideas presented in Fehd [10, sfg2014.1899].
- Contributors [4, www-wiki.dimensionless-quantity] provides the description of a ratio as a dimensionless quantity, a pure number.
- Contributors [1, www-unesco.types-of-variables],
Contributors [2, www.oswego.variable-types] and
Contributors [3, www.stattrek.what-are-variables] discuss the differences between continuous and discrete variables.

References

- [1] Various Contributors. Types of variables. In *Statistics*. UNESCO, 2001. URL http://www.unesco.org/webworld/idams/advguide/Chapt1_3.htm. continuous or quantitative, discrete or qualitative; online; accessed 2013-Mar-01.
- [2] Various Contributors. Variable types. In *Statistics*. SUNY Oswego, 2001. URL http://www.oswego.edu/~srp/stats/variable_types.htm. categorical, explanatory ordinal, quantitative, response; online; accessed 2013-Mar-01.
- [3] Various Contributors. What are variables. In *Descriptive Statistics*. Stat Trek, 2001. URL <http://stattrek.com/descriptive-statistics/variables.aspx>. qualitative or quantitative, continuous or discrete, univariate or bivariate; online; accessed 2013-Mar-01.
- [4] Wikipedia Contributors. Dimensionless quantity. In *Wikipedia, The Free Encyclopedia*, 2013. URL http://en.wikipedia.org/w/index.php?title=Dimensionless_quantity&oldid=539704643. properties, examples and list; online; accessed 2013-Mar-01.
- [5] Editor R.J. Fehd. Cardinality-ratio. In *sasCommunity.org*, 2008. URL http://www.sascommunity.org/wiki/Cardinality_Ratio. topics: definition and programs.

- [6] Editor R.J. Fehd. SmryEachVar: A data-review suite for each variable in all data sets in a libref. In *sasCommunity.org*, 2008. URL http://www.sascommunity.org/wiki/SmryEachVar_A_Data_Review_Suite. list processing using parameterized includes.
- [7] Ronald J. Fehd. SmryEachVar: A data-review routine for all data sets in a libref. In *SAS Global Forum Annual Conference Proceedings*, 2008. URL <http://www2.sas.com/proceedings/forum2008/003-2008.pdf>. Applications Development, 24 pp.; call execute, data review, data structure, dynamic programming, list processing, parameterized includes, utilities (writattr, writvalu) to repair missing elements in data structure; best contributed paper in ApDev.
- [8] Ronald J. Fehd. Database vocabulary: Is your data set a dimension (lookup) table, a fact table or a report? In *Western Users of SAS Software Annual Conference Proceedings*, 2008. URL <http://wuss.org/proceedings08/08WUSS%20Proceedings/papers/dmw/dmw04.pdf>. Databases and Warehouses, 8 pp.; topics: cardinality ratio, categories of columns (variables) and tables (data sets).
- [9] Ronald J. Fehd. Data review information: N-levels or cardinality ratio. In *SAS Global Forum Annual Conference Proceedings*, 2013. URL <http://support.sas.com/resources/papers/proceedings13/299-2013.pdf>. Statistics and Data Analysis, 6 pp.; using proc freq nlevels and nobobs to calculate cardinality ratio — range in (0:1) — of a variable to determine its type in (continuous, discrete, unique, worthless).
- [10] Ronald J. Fehd. Macro design ideas: Theory, template, practice. In *SAS Global Forum Annual Conference Proceedings*, 2014. URL <http://support.sas.com/resources/papers/proceedings14/1899-2014.pdf>. 21 pp.; topics: logic, quality assurance, testing, style guide, documentation, bibliography.
- [11] Ronald J. Fehd. List processing macro call-macro. In *Proceedings of the Western Users of SAS Software Annual Conference*, 2014. Coders Corner, 19 pp.; using %sysfunc with SCL functions to read a list, a control data set, and for each observation, call a macro with variable names and values as named parameters.

Closure

Contact Information:

Ronald J. Fehd

<mailto:Ron.Fehd.macro.maven@gmail.com>
http://www.sascommunity.org/wiki/Ronald_J._Fehd

About the author:

education:	B.S. Computer Science, U/Hawaii,	1986
	SAS User Group conference attendee since	1989
	SAS-L reader	since 1994
experience:	programmer: 25+ years	
	data manager using SAS:	17+ years
	statistical software help desk:	7+ years
	author: 30+ SUG papers	
	sasCommunity.org: 300+ pages	
SAS-L:	author: 6,000+ messages to SAS-L since	1997
	Most Valuable SAS-L contributor:	2001, 2003

Trademarks

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.
