

Sorting by formatted values

Jim Groeneveld, OCS Consulting, Rosmalen, the Netherlands.

A. ABSTRACT

Using PROC SORT in SAS® BY variables with associated formatted values just sorts the records according to their raw, unformatted values. If one explicitly wants to sort according to the (unique) formatted values one may use PROC FREQ instead, while involving all (desired) variables. However, care has to be taken if different unformatted values have equal formatted values. PROC FREQ, logically regards those as one category, with the occurring lowest unformatted value. E.g. if the unformatted values 1, 3, 5, 7 and 9 have the formatted value 'odd', and 2, 4, 6, 8 and 10 'even', then occurring values of 4, 6, 7 and 9 will be reduced to the categories 'even' and 'odd' with unformatted values of 4 and 7. Advantages and disadvantages, drawbacks and pitfalls of this kind of use of PROC FREQ, especially with equally formatted raw values, will be discussed.

B. INTRODUCTION

PROC FREQ and PROC REPORT may be used to produce aggregated statistics for variables with formatted values. Problems may arise if unformatted, non-consecutive values have identically formatted values, especially if BY variables are involved. Presorting according to the formatted values is necessary in such instances. PROC SORT can't do that, but it appears PROC FREQ can be used to account for formatted sorting while at the same time producing the aggregated statistics. Clearly some procedures explicitly ignore formatted values (in order to perform numeric arithmetic with unformatted values), like UNIVARIATE (even if only requesting frequencies and percentages), MEANS, etc.. Other procedures allow the use of (permanently or temporary) associated formats, like FREQ, TABULATE, REPORT, PRINT, etc.. Evidently PROC SORT ignores formats completely. The goal is to (pre)sort records (for REPORT) according to formatted classification values. Without creating new variables with the formatted contents (using function PUT) to sort by, it has appeared possible to sort by formatted values.

Here the features of PROC FREQ are being presented, which may combine formatted sorting with aggregation simultaneously.

C. FORMATTED VALUE PROCESSING

In order to study the effect of equally formatted nonconsecutive unformatted values on aggregating (class) variables with PROC FREQ consider the following example program:

```
OPTIONS FORMCHAR = '|----|||---' LINESIZE=78 PAGESIZE=66 FORMDLIM='=';
```

```
PROC FORMAT;
```

```
  VALUE _Number
```

```
    0, 2, 4, 6, 8 = "Even"
```

```
    1, 3, 5, 7, 9 = "Odd"
```

```
  ;
```

```
RUN;
```

```
DATA Test;
```

```
  Variable=9; OUTPUT;
```

```
  Variable=6; OUTPUT;
```

```
  Variable=4; OUTPUT;
```

```
  Variable=7; OUTPUT;
```

```
  FORMAT Variable _Number.;
```

```
RUN;
```

```
TITLE1 "Result 1a of PRINT with associated format";
```

```
TITLE2 "formatted values in same order as in original";
```

```
PROC PRINT DATA=Test; RUN;
```

```
TITLE1 "Result 1b of PRINT with dissociated format";
```

```
TITLE2 "unformatted values in same order as in original";
```

```
PROC PRINT DATA=Test; FORMAT Variable; RUN;
```

```
TITLE1 "Result 2a of FREQ with associated format";
```

```
TITLE2 "identically formatted values as one category: formatted value";
```

```
PROC FREQ DATA=Test; TABLE Variable / OUT=Aggr; RUN;
```

```
TITLE1 "Result 2b of PRINT of AGGR with associated format";
```

```
TITLE2 "identically formatted values as one category: formatted value";
```

```
PROC PRINT DATA=Aggr; RUN;
```

PhUSE 2006

```
TITLE1 "Result 2c of PRINT of AGGR with dissociated format";
TITLE2 "identically formatted values as ONE category: LOWEST UNformatted value";
PROC PRINT DATA=Aggr; FORMAT Variable; RUN;
```

And also consider its resulting (listing) output:

```
=====
      Result 1a of PRINT with associated format              1
      formatted values in same order as in original
                                     15:52 Wednesday, May 31, 2006
```

Obs	Variable
1	Odd
2	Even
3	Even
4	Odd

```
=====
      Result 1b of PRINT with dissociated format              2
      unformatted values in same order as in original
                                     15:52 Wednesday, May 31, 2006
```

Obs	Variable
1	9
2	6
3	4
4	7

```
=====
      Result 2a of FREQ with associated format                3
      identically formatted values as one category: formatted value
                                     15:52 Wednesday, May 31, 2006
```

The FREQ Procedure

Variable	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Even	2	50.00	2	50.00
Odd	2	50.00	4	100.00

```
=====
      Result 2b of PRINT of AGGR with associated format              4
      identically formatted values as one category: formatted value
                                     15:52 Wednesday, May 31, 2006
```

Obs	Variable	COUNT	PERCENT
1	Even	2	50
2	Odd	2	50

```
=====
      Result 2c of PRINT of AGGR with dissociated format              5
      identically formatted values as ONE category: LOWEST UNformatted value
                                     15:52 Wednesday, May 31, 2006
```

Obs	Variable	COUNT	PERCENT
1	4	2	50
2	7	2	50

With respect to PROC FREQ it appears that FREQ regards all raw, unformatted values, value lists or ranges with the same formatted value of TABLE variables as a single category with a raw value equal to the lowest occurring value for that category and associates its format to the resulting aggregated variable in its output dataset. The same appears to apply to PROC REPORT. Thus both procedures correctly generate an output dataset based on formatted values (of TABLE or COLUMN variables), represented by only one original raw value per formatted value for each variable in the (aggregated) output dataset.

D. BY VARIABLE'S FORMATTED VALUES PROCESSING

BY variables (with possibly any procedure) are commonly sorted according to their `_unformatted_` values (using PROC SORT) and BY-level groups may be formed according to the `_formatted_` values if BY variables have associated formats with any procedure. There are 3 distinctive format correspondence situations:

1. each occurring raw, unformatted value has a unique complementary formatted value;
2. a formatted value is associated with a range of succeeding, discrete or continuous, unformatted values;
3. several (quite) different unformatted values or ranges are associated with the same formatted value, while there are other, intermediate, unformatted values associated with other formatted values (as in the example).

The SAS (9) documentation (`base_proc_8417.pdf`) says about formatted BY variables:

"When a procedure processes a data set, it checks to see if a format is assigned to the BY variable. If it is, then the procedure adds observations to the current BY groups until the formatted value changes. If nonconsecutive internal values of the BY variable(s) have the same formatted value, then the values are grouped into different BY groups. This results in two BY groups with the same formatted value. Further, if different and consecutive internal values of the BY variable(s) have the same formatted value, then they are included in the same BY group."

This clearly is not the kind of accounting for formatted BY variables that is wanted here.

If the data are sorted correctly, i.e. by the raw values, BY-variable processing is performed without SAS errors in all situations, though (some) REPORT results may be incorrect in situation 2 due to unwanted summing. In situation 3, however, this leads, to multiple occasions of the same BY-group (BY-variables value combination) because of the format with which quite different, not succeeding values have the same formatted representation. This situation would need an alternative preceding sort in order to avoid multiple occasions of the same BY-group. This sorting algorithm could be described as:

1. forming groups of unformatted values with the same formatted value, assigning them the lowest group value;
 2. within each group sort according to the original, unformatted values;
 3. sort the groups according to the new group variables with their lowest unformatted value.
- This is quite different from, more complicated than and better than a sort according to formatted values only, which would generally not be good enough for BY-variables and might generate a SAS error, as the dataset still might not be sorted properly according to unformatted values.

E. SORTING BY FORMATTED BY VARIABLES DURING AGGREGATION

If it is desired to run PROC FREQ to calculate frequencies and percentages (aggregated statistics), which finally are being shown with PROC REPORT, a sorting algorithm should support all kinds of formatted BY-variables as well. So some complex combination of unformatted and formatted sort for the BY-variables as described above is needed while it is not desirable to specify formats explicitly: permanently associated formats should be applied implicitly. There are two ways to accomplish this:

1. instead of indicating the BY-variables as such with PROC FREQ, they might be involved as (initial) TABLE variables as well. Presorting is not necessary then and the same frequencies and percentages are being output. This step eliminates multiple occurrences of formatted values combinations and the output dataset has (default) been sorted according to the unformatted values of all involved variables in the order of TABLE specification, where original raw values with non-unique formatted values have automatically changed into the lowest occurring corresponding raw value.

(Subsequently PROC SORT might be run optionally to sort the unique categories according to their resulting unformatted, lowest values (from PROC FREQ) in some different order (e.g. descending or a different hierarchy of the variables) or to resort the appended output datasets from multiple calls to PROC FREQ in the same order. Finally PROC REPORT would present all of it, while all variables still have their original format permanently associated.)

2. an extra, preceding PROC FREQ might do the job of sorting, where the BY-variables (to be used as such with the actual PROC FREQ and PROC REPORT) are not included as BY-variables, but as (initial) TABLE variables as well (just like alternative A). The preceding PROC FREQ generates an aggregated output dataset sorted according to all involved variables (incl. those BY-variables). This is sufficient in case one can continue with an aggregated dataset with specific variables only, that must be used in the actual, subsequent PROC FREQ with the "WEIGHT Count;" statement. In this instance the first PROC FREQ would only determine the COUNTs for the distinctive category combinations, while the actual, second PROC FREQ would calculate those too, but several desired percentages and other statistics as well. This second PROC FREQ then actually includes the BY-variables as BY-variables.

Both methods may differ in the amount of data they yield: method A yields aggregated data for all theoretically possible value combinations of the BY-variables, even where some combination has no observations, and of all existing values (combinations) for the other variables over the whole table for each BY-group, if option SPARSE is specified. Method B finally only yields aggregated data for all occurring value combinations of the BY-variables, and existing values (combinations) for the other variables within a particular BY-group, if option SPARSE is specified with the second PROC FREQ.

The first PROC FREQ should never have the SPARSE option turned on at all: it is not necessary, but it also generates a SAS bug (in SAS vs. 6.12). The bug stops a (batch) run with a popup error window and the log output:

```
NOTE: By group contains 0 observations with a nonzero weight.
ERROR: Unknown exception (80000602)
ERROR: A severe error occurred in task FREQ for module UNKNOWN executing
```

PhUSE 2006

in module UNKNOWN at address 00A64A0F.

Please contact Technical Support to report this error.

ERROR: Generic critical error.

NOTE: The SAS System stopped processing this step because of errors.

This problem occurs in the subsequent PROC FREQ after keeping BY-variable value combinations with frequencies of 0 (using the SPARSE option) in the initial PROC FREQ. They should be discarded before running a next PROC FREQ with the "WEIGHT Count;" statement. The ERROR does not occur with SAS vs. 9.1.3, the NOTE remains; I don't know about SAS vs. 8.2. And the remark in the NOTE is actually incorrect: there is an observation (or even more) with a zero weight, not zero observations with a nonzero weight.

Method B, the preceding alternative sort using FREQ, seems the best one in this case. It can be viewed as an alternative for some initial formatted sort. All involved variables, needed later, have to be specified. The resulting dataset is aggregated regarding both the usual variables as the original BY-variables. Multiple occurrences of formatted values (corresponding to more than one different unformatted values), or combinations of them, are reduced to only one (unformatted) value (combination). Using COUNT subsequently as the weight makes this method suited for preprocessing for subsequent procedures other than FREQ as well, e.g. TABULATE, UNIVARIATE, MEANS, REPORT. It has the additional advantage (or disadvantage if you view it that way) of yielding a smaller dataset than the original one. It does not concern a rearrangement, another ordering of records, but it only concerns records, that are aggregated according to formatted values, containing a specified subset of variables. One has to keep in mind that these data may not yet be sorted properly according to (corresponding lowest) unformatted values.

A real ordering according to formatted values might be accomplished using PROC FREQ too, by specification of the option ORDER=FORMATTED in the procedure call, resulting in an aggregated dataset with only the specified variables. PROC FREQ, PROC TABULATE, PROC MEANS, and probably several other procedures, have the same ORDER option as PROC REPORT, though while PROC REPORT allows the option to be specified per individual variable, the other procedures only allow specification of the option globally, for all variables at a time.

F. ALTERNATIVES

SAS offers alternatives to sort a dataset according to formatted values. Examples are PROC REPORT's own sorting mechanism (to force formatted sorting of table values), while aggregating or not, and creating (PUT function) new variable(s) with formatted values in a view or new (temporary, not aggregated) dataset to sort by, but as that may yield (much) larger datasets or views it should rather be avoided.

Another solution, applying PROC SQL is:

```
PROC SQL;  
  CREATE TABLE Sorted AS SELECT * FROM Unsorted ORDER BY PUT(SortVar, Formated.);  
RUN; QUIT;
```

This yields a sorted (according to formatted values), not aggregated dataset that has to be processed further to produce aggregated statistics (PROC FREQ or REPORT).

One has to keep in mind that such a sorted dataset still contains all different unformatted values associated with the (equally) formatted ones. And such a dataset might not at all be sorted properly yet according to the various unformatted values of the concerning variables. Hence the use of such a sorted variable still doesn't make it always suited for being used as a BY variable; the NOTSORTED option then is very much needed. Generally, and always with PROCs, a BY-group changes when both its unformatted and formatted value changes. With data steps one has to specify the GROUPFORMAT option in the BY statement to force grouping according to formatted values of a BY variable, where grouping according to the unformatted values is the default.

G. CONCLUSION

Actually sorting according to formatted values makes most sense if aggregating (and limiting to certain variables) at the same time. Not only two or more things can be done at the same time, but the main advantage is that the unformatted values associated with the formatted ones have been reduced to one per formatted value. This unformatted, raw value thus is uniquely associated with the formatted one and is the lowest one of the original, unformatted values associated with the formatted one. One should still bear in mind that a dataset sorted like this may not necessarily be sorted properly according to the unformatted values of the concerning variable.

REFERENCES

SAS Institute Inc. 2004. Base SAS® 9.1.3 Procedures Guide. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

Y. (Jim) Groeneveld
OCS Consulting Benelux
PO BOX 490
5240 AL ROSMALEN
THE NETHERLANDS
Office: +31 (0)73 523 6000
Fax.: +31 (0)73 523 6600
JimG@OCS-Consulting.com
www.ocs-consulting.com
home.hccnet.nl/jim.groeneveld