PharmaSUG 2015 - Paper QT14

Getting the Most Out of PROC SORT: A Review of Its Advanced Options

Max Cherny, GlaxoSmithKline, King of Prussia, PA

ABSTRACT

The paper describes the use of some underutilized, yet extremely useful, options of PROC SORT. These options include SORTSEQ, CASE_FIRST, NUMERIC_COLLATION, DUPOUT, NOUNIQUEKEY and others. The paper provides easy and reusable examples for each of the options.

INTRODUCTION

PROC SORT is one of the most commonly used procedures used by SAS users. PROC SORT is used to sort a SAS data set by a variable or a set of variables in order to prepare the data set for subsequent use in a data step or a procedure. The basic and the most commonly used options of PROC SORT are OUT and NODUPKEY. Most SAS users are familiar with these options and they are often implemented in SAS programs.

However PROC SORT also provides a number of other underused options which can save a great deal of effort. These options help define very specific rules by which a data set can be manipulated.

LINGUISTIC

The LINGUISTIC option is a very powerful tool to sort alphanumeric data based on various requirements.

Let's consider the following data set:

```
data ae;
  input aeterm $1-30 ;
  datalines;
Cough
Fall
fall
cough
;
run;
```

AETERM
Cough
Fall
fall
cough

Table 1. AE data set

Using PROC SORT by default results in sorting the data first by upper case followed by the lower case:

```
proc sort data=ae ;
  by aeterm;
run;
```

AETERM
Cough
Fall
cough
fall

Table 2. AE data set sorted with the default options

However, there is a way to sort the entire data alphabetically ignoring the case of the letters:

```
proc sort data=ae SORTSEQ =LINGUISTIC ;
  by aeterm;
run;
```

AETERM
cough
Cough
fall
Fall

Table 3. AE data set sorted with SORTSEQ =LINGUISTIC option

The LINGUISTIC option can be used to sort data according to various rules. For example, if required, CASE_FIRST =UPPER rule will allow to sort the data alphabetically by upper case first, and then by lower case.

```
proc sort data=ae SORTSEQ =LINGUISTIC (CASE_FIRST=UPPER);
  by aeterm;
run;
```

AETERM
Cough
cough
Fall
fall

Table 4. AE data set sorted with SORTSEQ =LINGUISTIC (CASE_FIRST=UPPER) option

NUMERIC_COLLATION

Some clinical trial data is recorded in character form such as visits or subject IDs. One way to sort such data is to create a corresponding numeric code, and then sort the data by that field. However the NUMERIC_COLLATION option allows SAS users to sort character fields if such fields contain numbers.

The simple program below creates a data set containing VISIT variable.

```
data visits;
  input visit $1-15 ;
  datalines;
 Day 22
 Day 3
 Day 2
;
run;
```

VISIT
Day 22
Day 3
Day 2

Table 5. VISITS data set

The following code will sort the data alphabetically:

```
proc sort data= visits ;
  by visit;
run;
```

VISIT
Day 2
Day 22
Day 3

Table 6. VISITS data set sorted by VISIT

The data set is not sorted numerically since Day 2 is followed by the Day 22. Normally a SAS user would need a numeric code variable to sort the visit days in the correct order. However, with the NUMERIC_COLLATION =ON option it is possible to sort the VISIT data set in numeric order:

```
proc sort data= visits SORTSEQ =LINGUISTIC (NUMERIC_COLLATION=ON);
  by visit;
run;
```

VISIT
Day 2
Day 3
Day 22

Table 7. VISITS data set sorted with SORTSEQ =LINGUISTIC (NUMERIC_COLLATION=on) option

The updated data set now has VISIT variables sorted in numeric order even if the variables are characters.

ALTERNATE_HANDLING

The ALTERNATE_HANDLING option can be useful when it is necessary to control sorting of special characters such as spaces.

The following data set contains names of investigators in a study:

```
data investig;
  input invname $1-15 ;
  datalines;
 john smith1
 johnsmith4
 john smith3
 johnsmith2
;
run;
```

INVNAME
john smith1
johnsmith4
john smith3
johnsmith2

Table 8. INVESTIG data set

By default PROC SORT will sort this data by taking into account the blank space between first and last names.

```
proc sort data=investig ;
  by invname;
run;
```

INVNAME
john smith1
john smith3
johnsmith2
johnsmith4

Table 9. INVESTIG data set sorted by INVNAME

However, adding the ALTERNATE_HANDLING rule to PROC SORT will ignore the blank characters in the data.

```
proc sort data=investig SORTSEQ =LINGUISTIC (ALTERNATE_HANDLING=SHIFTED);
  by invname;
run;
```

INVNAME
john smith1
johnsmith2
john smith3
johnsmith4

Table 10. INVESTIG data set sorted with SORTSEQ =LINGUISTIC (ALTERNATE_HANDLING=SHIFTED)

SORTING UNIQUE OR DUPLICATE OBSERVATIONS

PROC SORT provides a number of options to analyze a data set containing duplicate or unique observations. The NOUNIQUEKEY option is new starting with SAS 9.3. It is useful to identify duplicate observations. This option is somewhat opposite of the NODUPKEY option which removes duplicate observations. The following example will demonstrate the difference:

```
data advs;
  infile datalines delimiter=',';
  input subjid $  visit $ param $ aval ;
  datalines;
 1, week 1, HR, 78
 1, week 1, HR, 78
 1, week 2, HR, 79
 1, week 2, HR, 79
 1, week 3, HR, 77
 2, week 1, HR, 80
;
run;
```

SUBJID	VISIT	PARAM	AVAL
1	week 1	HR	78
1	week 1	HR	78
1	week 2	HR	79
1	week 2	HR	79
1	week 3	HR	77
2	week 1	HR	80

Table 11. ADVS dataset

There are duplicate observations at week 1 and week 2 which can be removed with NODUPKEY option.

```
proc sort data=advs NODUPKEY;
  by subjid visit aval;
run;
```

SUBJID	VISIT	PARAM	AVAL
1	week 1	HR	78
1	week 2	HR	79
1	week 3	HR	77
2	week 1	HR	80

Table 12. ADVS dataset sorted with NODUPKEY option

However, NOUNIQUEKEY option will write duplicate observations and remove any observations which are unique:

```
proc sort data=advs NOUNIQUEKEY ;
  by subjid visit aval;
run;
```

SUBJID	VISIT	PARAM	AVAL
1	week 1	HR	78
1	week 1	HR	78
1	week 2	HR	79
1	week 2	HR	79

Table 13. ADVS dataset sorted with NOUNIQUEKEY option

In order to see which observations were removed with NOUNIQUEKEY, the UNIQUEOUT option can be used.

```
proc sort data=advs NOUNIQUEKEY UNIQUEOUT=observations_removed ;
  by subjid visit aval;
run;
```

SUBJID	VISIT	PARAM	AVAL
1	week 3	HR	77
2	week 1	HR	80

Table 14. Data set created with UNIQUEOUT option

Note that the resulting data set does not contain all unique observations from the ADVS data set. It only contains the observations removed by NOUNIQUEKEY option. The NOUNIQUEKEY and UNIQUEOUT options must be used together.

PROC SORT also has the DUPOUT option to specify the data set to which all observations deleted by NODUPKEY are written.

```
proc sort data=advs NODUPKEY DUPOUT=observations_removed;
  by subjid visit aval;
run;
```

SUBJID	VISIT	PARAM	AVAL
1	week 1	HR	78
1	week 2	HR	79

Table 15. Data set created with DUPOUT option

CONCLUSION

PROC SORT provides many useful options to manage duplicate observations or sort alphanumeric characters. The author suggests exploring the additional rules in SORTSEQ option described in SAS manual. These options might save SAS users a significant amount of time.

PROC SORT options/rules	Purpose
SORTSEQ	general option to control sorting of alphanumeric characters
LINGUISTIC	sorts characters according to rules of the specified language
CASE_FIRST	specifies the order of uppercase and lowercase letters
NUMERIC_COLLATION	orders integer values within the text by the numeric value
ALTERNATE_HANDLING	controls the handling of special characters
NODUPKEY	removes duplicate records
NOUNIQUEKEY	removes unique records
UNIQUEOUT	specifies the output data set for observations eliminated by the NOUNIQUEKEY option
DUPOUT	specifies the output data set for observations eliminated by the NODUPKEY option

Table 16. PROC SORT options described in the paper

REFERENCES

Wright, W. Checking for Duplicates, SUGI, 2007.Available at http://www2.sas.com/proceedings/sugi31/249-31.pdf.

Mebust and Bridger, Creating Order out of Character Chaos: Collation Capabilities of the SAS System, SUGI, 2007. Available at http://www2.sas.com/proceedings/sugi31/249-31.pdf.

ACKNOWLEDGMENTS

Author would like to thank Greg Cicconetti, Ph.D. for his help with this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Max Cherny
GlaxoSmithKline
Email: chernym@yahoo.com