Paper 037-30

The Mystery of the PROC SORT Options NODUPRECS and NODUPKEY Revealed

Britta Kelsey, MGI Pharma, Inc., Bloomington, MN

ABSTRACT

The NODUPRECS (or NODUP) and NODUPKEY options can be useful with the SORT procedure but they can be dangerous if you do not understand them completely. They work similarly in that they both can eliminate unwanted observations, but NODUPRECS compares all the variables in your data set while NODUPKEY compares just the BY variables. Also, you must be aware of how your data set is currently sorted to eliminate the observations that you want because these options compare adjacent observations. In this paper, I will describe this in greater detail and show examples of how to use the NODUPRECS and NODUPKEY options. All examples shown were done in the SAS® system for PCs, version 8.2. The intended audience for this paper is beginner level SAS programmers.

INTRODUCTION

There seems to be confusion among SAS users about the NODUPRECS (or NODUP) and NODUPKEY options. These can be very useful but dangerous if you are not clear on how they work. I will bring some clarification to these PROC SORT options and will show some examples of how they work so you can put them to good use. Throughout the rest of this paper I will be using the alias NODUP for the NODUPRECS option.

DEFINING NODUP AND NODUPKEY OPTIONS

The NODUP option checks for and eliminates duplicate observations. If you specify this option, PROC SORT compares all variable values for each observation to those for the previous observation that was written to the output data set. If an exact match is found, the observation is not written to the output data set.

The NODUPKEY option checks for and eliminates observations with duplicate BY variable values. If you specify this option, PROC SORT compares all BY variable values for each observation to those for the previous observation written to the output data set. If an exact match using the BY variable values is found, the observation is not written to the output data set.

Notice that with the NODUPKEY option, PROC SORT is comparing all *BY* variable values while the NODUP option compares *all* the variables in the data set that is being sorted. An easy way to remember the difference between these options is to keep in mind the word “key” in NODUPKEY. It evaluates the “key” or BY variable values that you specify. One thing to beware of with both options is that they both compare the previous observation written to the output data set. So, if the observations that you want eliminated are not adjacent in the data set after the sort, they will not be eliminated.

NODUP EXAMPLES

Below is the code for the data set, called BEST, that I will be using throughout this paper. This data set is similar to one used for a clinical study and contains patient numbers, the study arm they are assigned to, their best response during the study, and the number of days their medication was delayed. Notice that patient 01 has two observations that are exactly the same and patient 03 has two observations with only the variable BESTRES being different.

```
data best;
  input patient 1-2 arm $ 4-5 bestres $ 6-7 delay 9-10;
       datalines;
       01 A CR 0
       02 A PD 1
       03 B PR 1
```

```
            04 B CR 2
            05 C SD 1
            06 C SD 3
            07 C PD 2
            01 A CR 0
            03 B PD 1
            ;
      run;
```

Example 1:

In this example the SORT procedure is used with the NODUP option. By using a PROC SORT, I want to order the data by the variable PATIENT and eliminate any observations that have the exact same information for all variables.

```
      proc sort data=best nodup out=ex1;
            by patient;
      run;
```

Here is how the output data set EX1 looks:

```
            PATIENT  ARM       BESTRES  DELAY
            01       A         CR       0
            02       A         PD       1
            03       B         PR       1
            03       B         PD       1
            04       B         CR       2
            05       C         SD       1
            06       C         SD       3
            07       C         PD       2
```

Notice that patient 03 still has two observations because all the variable values are not the same for these two observations.

Example 2:

In this example, I want to order the data by ARM and eliminate any observations that have the exact same information for all variables. Using the PROC SORT statement, I first use the BY variable ARM to see if I get the result I want.

```
      proc sort data=best nodup out=ex2_1;
            by arm;
      run;
```

Here is how the output data set EX2_1 looks:

```
            PATIENT  ARM       BESTRES  DELAY
            01       A         CR       0
            02       A         PD       1
            01       A         CR       0
            03       B         PR       1
            04       B         CR       2
            03       B         PD       1
            05       C         SD       1
            06       C         SD       3
            07       C         PD       2
```

This sort did not do the elimination that I wanted. This is because when the procedure sorted by ARM, the second observation for patient 01 just moved up under patient 02. The two observations for patient 01 are not adjacent; therefore, the second one did not get eliminated. To get the results that I want, I will have to sort by the ARM and PATIENT variables.

```
      proc sort data=best nodup out=ex2_2;
            by arm patient;
```

```
        run;
```

Here is how the output data set EX2_2 looks:

```
PATIENT   ARM       BESTRES   DELAY
01        A         CR        0
02        A         PD        1
03        B         PR        1
03        B         PD        1
04        B         CR        2
05        C         SD        1
06        C         SD        3
07        C         PD        2
```

Now the second observation for patient 01 is eliminated and the data set is ordered by arm which is the desired result.

NODUPKEY EXAMPLES

For these examples, I will be using the data set BEST from above.

Example 3:

In this example the SORT procedure is used with the NODUPKEY option. This is similar to example 1 above but I am going to use the NODUPKEY option instead of the NODUP option and compare the difference in results.

```
        proc sort data=best nodupkey out=ex3;
              by patient;
        run;
```

Here is how the output data set EX3 looks:

```
PATIENT   ARM       BESTRES   DELAY
01        A         CR        0
02        A         PD        1
03        B         PR        1
04        B         CR        2
05        C         SD        1
06        C         SD        3
07        C         PD        2
```

Notice that the PROC SORT eliminated one of the observations for patient 01 and one for patient 03. Because I sorted by the PATIENT variable and used the NODUPKEY option, the SORT procedure is checking for adjacent observations with only the same patient number and eliminating any duplicates after the first occurrence of that number. In contrast, in the output data set for example 1, patient 03 still had 2 observations.

Here it is important to note that the first observation for patient 03 has a PR for the variable BESTRES and the second observation for patient 03 has a PD for that variable. The NODUPKEY option is only looking at the BY variable of PATIENT so it ignores the difference in all other variables and it eliminates the second observation with a BESTRES of PD. This is where this option can be dangerous because you may eliminate observations that you actually wanted to keep since they may have different values for a non-BY variable.

Example 4:

In this example I will sort by the variable ARM as I did in example 2 above but instead of using the NODUP option, will use the NODUPKEY option to see the difference in results.

```
        proc sort data=best nodupkey out=ex4;
              by arm;
        run;
```

Here is how the output data set EX4 looks:

PATIENT	ARM	BESTRES	DELAY
01	A	CR	0
03	B	PR	1
05	C	SD	1

Since I sorted by the variable ARM and used the NODUPKEY option in this example, PROC SORT only kept the first observation it encountered for each arm and eliminated the duplicates after that. By making the one change in options from NODUP to NODUPKEY, the output data set EX4, with only one observation per arm, looks quite different from the output data set EX2_1 generated in example 2.

Example 5:

In this example I will sort by the variables ARM and BESTRES with the NODUPKEY option to see what will happen to my data set.

```
       proc sort data=best nodupkey out=ex5;
              by arm bestres;
       run;
```

To help better understand what is being eliminated, I will once again show the data set BEST and compare it to the output data set EX5:

Data set BEST -

PATIENT	ARM	BESTRES	DELAY
01	**A**	**CR**	**0**
02	A	PD	1
03	B	PR	1
04	B	CR	2
05	**C**	**SD**	**1**
06	**C**	**SD**	**3**
07	C	PD	2
01	**A**	**CR**	**0**
03	B	PD	1

Data set EX5 –

PATIENT	ARM	BESTRES	DELAY
01	**A**	**CR**	**0**
02	A	PD	1
04	B	CR	2
03	B	PD	1
03	B	PR	1
07	C	PD	2
05	**C**	**SD**	**1**

Here the second observation for patient 01 is eliminated because it has the same BY variable values as the first observation for patient 01 (patient 01 is colored coded in red). Also, patient 06 is eliminated because it has the same BY variable values as patient 05 (these patients are colored coded in blue).

CONCLUSION

The NODUP option in the SORT procedure eliminates observations that are exactly the same across *all* variables. The NODUPKEY option eliminates observations that are exactly the same across the *BY* variables. Keep in mind that both of these options compare adjacent observations in the output data set. Now that you know how the NODUP and NODUPKEY options work, you can use them in confidence to get the data set you want!

REFERENCES

SAS Institute Inc. (1990), *SAS Procedures Guide, Version 6, Third Edition,* Cary, NC: SAS Institute Inc.

ACKNOWLEDGEMENTS

I would like to thank Sujata Arora, Scott Lunos, and Scott McKane for their review and helpful comments on this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please feel free to contact the author at:

Britta Kelsey
MGI Pharma, Inc.
5775 West Old Shakopee Road, Suite 100
Bloomington, MN 55437-3174
E-mail: britta.kelsey@mgipharma.com