PharmaSUG 2016 – Paper TT01

Removing Duplicates Using SAS®

Kirk Paul Lafler, Software Intelligence Corporation, Spring Valley, California

Abstract

We live in a world of data – small data, big data, and data in every conceivable size between small and big. In today's world data finds its way into our lives wherever we are. We talk about data, create data, read data, transmit data, receive data, and save data constantly during any given hour in a day, and we still want and need more. So, we collect even more data at work, in meetings, at home, using our smartphones, in emails, in voice messages, sifting through financial reports, analyzing profits and losses, watching streaming videos, playing computer games, comparing sports teams and favorite players, and countless other ways. Data is growing and being collected at such astounding rates all in the hopes of being able to better understand the world around us. As SAS professionals, the world of data offers many new and exciting opportunities, but also presents a frightening realization that data sources may very well contain a host of integrity issues that need to be resolved first. This presentation describes the available methods that are used to remove duplicate observations (or rows) from data sets (or tables) based on the row's values and/or keys using SAS®.

Introduction

An issue found in some data sets is the presence of duplicate rows and/or duplicate keys. When found, SAS can be used to remove any unwanted data. **Note:** Before duplicates are removed, be sure to consult with your organization's data analyst or subject matter expert to see if removal is necessary or permitted. It's better to be safe than sorry. This paper will illustrate two very different approaches to remove duplicate observations (or rows) from data sets (or tables) based on the row's values and/or keys using SAS®. Each example is illustrated using a single data set, MOVIES. The Movies data set contains 26 rows, and has a structure consisting of six columns. Title, Category, Studio, and Rating are defined as character columns; and Length and Year are defined as numeric columns. The Movies data set contains two duplicate rows – Brave Heart and Rocky; and two duplicate Title keys – Forrest Gump and The Wizard of Oz, shown below.

	Title	Length	Category	Year	Studio	Rating
1	Brave Heart	177	Action Adventure	1995	Paramount Pictures	R
2	Casablanca	103	Drama	1942	MGM / UA	PG
3	Christmas Vacation	97	Comedy	1989	Warner Brothers	PG-13
4	Coming to America	116	Comedy	1988	Paramount Pictures	R
5	Dracula	130	Horror	1993	Columbia TriStar	R
6	Dressed to Kill	105	Drama Mysteries	1980	Filmways Pictures	R
7	Forrest Gump	142	Drama	1994	Paramount Pictures	PG-13
8	Ghost	127	Drama Romance	1990	Paramount Pictures	PG-13
9	Jaws	125	Action Adventure	1975	Universal Studios	PG
10	Jurassic Park	127	Action	1993	Universal Pictures	PG-13
11	Lethal Weapon	110	Action Cops & Robber	1987	Warner Brothers	R
12	Michael	106	Drama	1997	Warner Brothers	PG-13
13	National Lampoon's Vacation	98	Comedy	1983	Warner Brothers	PG-13
14	Poltergeist	115	Horror	1982	MGM / UA	PG
15	Rocky	120	Action Adventure	1976	MGM / UA	PG
16	Scarface	170	Action Cops & Robber	1983	Universal Studios	R
17	Silence of the Lambs	118	Drama Suspense	1991	Orion	R
18	Star Wars	124	Action Sci-Fi	1977	Lucas Film Ltd	PG
19	The Hunt for Red October	135	Action Adventure	1989	Paramount Pictures	PG
20	The Terminator	108	Action Sci-Fi	1984	Live Entertainment	R
21	The Wizard of Oz	101	Adventure	1939	MGM / UA	G
22	Titanic	194	Drama Romance	1997	Paramount Pictures	PG-13
23	Rocky	120	Action Adventure	1976	MGM / UA	PG
24	Brave Heart	177	Action Adventure	1995	Paramount Pictures	R
25	Forrest Gump	143	Drama	1994	Paramount Pictures	PG-13
26	The Wizard of Oz	102	Adventure	1939	MGM / UA	G

Method #1 – Using PROC SORT

The first method, and one that is popular with SAS professionals everywhere, uses PROC SORT to remove duplicates. The SORT procedure supports three options for the removal of duplicates: NODUPRECS, NODUPKEYS, and DUPOUT=.

The NODUPRECS (or NODUP) Option

By specifying the NODUPRECS (or NODUPREC) (or NODUP) option with PROC SORT, rows with identical values for all columns are removed from the output data set. The resulting output data saw the removal of the duplicate rows for Brave Heart and Rocky because they have identical data for all columns.

```
PROC SORT DATA=Movies
          OUT=Movies_Sorted_NoDuprecs
         NODUPRECS ;
  BY Title ;
RUN ;
```

The NODUPKEYS (or NODUPKEY) Option

By specifying the NODUPKEYS (or NODUPKEY) option with PROC SORT, rows with duplicate keys are automatically removed from the output data set. The resulting output data set saw the removal of all the duplicate rows for Brave Heart, Forrest Gump, Rocky and The Wizard of Oz because they have duplicate keys data for the column, Title.

```
PROC SORT DATA=Movies
          OUT=Movies_Sorted_NoDupkeys
         NODUPKEYS ;
  BY Title ;
RUN ;
```

The DUPOUT= Option

A DUPOUT= option is specified with PROC SORT to identify duplicate rows before actually removing them from a data set. The DUPOUT= option is used with either the NODUPKEYS or NODUPRECS option to name a data set that will contain duplicate keys or duplicate rows. The DUPOUT= option is generally used when the data set is too large for visual inspection. In the next code example, the DUPOUT= and NODUPKEY options are specified. The resulting output data set contains the duplicate rows for Brave Heart, Forrest Gump, Rocky and The Wizard of Oz.

```
PROC SORT DATA=Movies
        DUPOUT=Movies_Sorted_Dupout_NoDupkey
        NODUPKEY ;
  BY Title ;
RUN ;
```

In the next example, the DUPOUT= and NODUPRECS options are specified. The resulting output data set contains the duplicate rows for Brave Heart and Rocky because these rows have identical data for all columns.

```
PROC SORT DATA=Movies
        DUPOUT=Movies_Sorted_Dupout_NoDuprecs
        NODUPRECS ;
  BY Title ;
RUN ;
```

Note: Although the removal of duplicates using PROC SORT is popular with many SAS professionals, an element of care should be given to using this method when processing big data sets. Because sort operations are time consuming and CPU-intensive operations, requiring as much as three times the amount of space to sort a data set, excessive

demand is placed on system resources. Instead, SAS professionals may want to consider using PROC SUMMARY with the CLASS statement to avoid the need for sorting altogether, see Method #2.

Method #2 – Using PROC SUMMARY with the CLASS Statement

The second method of removing duplicates uses PROC SUMMARY with the CLASS statement. Using PROC SUMMARY with the CLASS statement provides SAS professionals with a more efficient alternative than PROC SORT, and other methods, by avoiding the need for sorting in advance. Without the sorting requirement, considerably less system resources are needed to identify duplicates. But two additional aspects make this method effective: the specification of a CLASS statement to collapse rows with the same column values and the creation of a _FREQ_ column containing the number of occurrences. As shown in the example, a WHERE statement, WHERE= data set option, or SQL WHERE-clause is specified to select rows with multiple occurrences (duplicates) with the WHERE-clause expression Dupkey > 1.

```
PROC SUMMARY DATA=Mydata.Movies_dups2 NWAY ;
  CLASS Title ;
  OUTPUT OUT=Movies_Summary_NoDupkey(DROP=_type_) ;
RUN ;
PROC PRINT DATA=Movies_Summary_NoDupkey(RENAME=(_FREQ_ = Dupkey)) NOOBS ;
  WHERE Dupkey > 1 ;
RUN ;
```

Conclusion

While many users use PROC SORT to remove duplicate observations or rows from SAS data sets, using PROC SUMMARY with the CLASS statement provides a more efficient alternative. Because sorts can be expensive and time-consuming processes, it's advisable to use approaches that reduce the utilization of system resources to remove duplicates, such as with PROC SUMMARY.

Acknowledgments

The author thanks Jacques Lanoue and William E. Benjamin Jr., Techniques & Tutorials Section Chairs for accepting my abstract and paper; Eric Larson, PharmaSUG 2016 Academic Chair; Sandra Minjoe, PharmaSUG 2016 Operations Chair; SAS Institute Inc.; and the PharmaSUG Executive Committee for organizing a great conference!

Trademark Citations

About the Author

Kirk Paul Lafler has used SAS since 1979, and is consultant and founder of Software Intelligence Corporation. He is a SAS Certified Professional, provider of IT consulting services, trainer to SAS users around the world, UCSD Extension professor, mentor, and sasCommunity.org emeritus Advisory Board member. As the author of six books including Google® Search Complete! (Odyssey Press. 2014) and PROC SQL: Beyond the Basics Using SAS, Second Edition (SAS Press. 2013); Kirk has written more than five hundred papers and articles; been an Invited speaker and trainer at five hundred-plus SAS International, regional, special-interest, local, and in-house user group conferences and meetings; and is the recipient of 23 "Best" contributed paper, hands-on workshop (HOW), and poster awards.

Comments and suggestions can be sent to:

Kirk Paul Lafler
Senior SAS® Consultant, Application Developer, Data Scientist, Educator and Author
Software Intelligence Corporation
E-mail: KirkLafler@cs.com
LinkedIn: http://www.linkedin.com/in/KirkPaulLafler
Twitter: @sasNerd