Abstract

Duplicated data, records, and other information is a pain. Duplicated information takes up space, can cause inaccurate measures, and cost money that can be used for something else. Finding those duplicated values in data, records, and other sources is easy using SAS' "FIRST." and "LAST." Expressions. This paper will explain how SAS can be used to help find those dups.

Introduction

Whenever a SAS dataset is sorted, the BY variables are assigned "FIRST."/"LAST." expressions that represent a single numeric value that you can use in a SAS program to see if that variable is the first and/or last in the sorted set. These expressions are either '1' for 'true' or '0' for 'false'. So to see if a variable is the only one, it should be the first and the last. This paper will illustrate the use of these expressions.

About SAS Expressions

- An 'expression' represents a single value that is used in a program.
- SAS expressions can be a constant, variable, or array element
- Expressions can combine constants and variables with functions calls and operators

Coding FIRST. LAST. To Find Duplicates

- Variables created when sorting containing a Boolean for each BY variable
- 1' if current observation is 'FIRST./LAST.' in the 'BY' group
- "0' if current observation is not 'FIRST./LAST.' in the 'BY' group

Before getting into the coding, let me layout the need for this program. VFW Post 6796, a veteran services organization that I am a member of, had a need to reduce the cost of producing the newsletter sent every other month to the 1500 members. After changing the layout to reduce the paper cost and make the postal service happy, a manual check of the mailing list showed some duplicated mailings. There wasn't enough time to manually check for duplicates, get the new spreadsheet ready for the printer, and get to the mail by the upcoming deadline. So I got the membership mailing spreadsheet and ran it thru SAS.

This paper will use the mailing list data to find duplicates using "FIRST."/"LAST." expressions. The code sample assumes that we have a file with duplicated names. After reading in the file, we'll sort it by the variable 'Full_Name', create a DATA set by SETting the sorted file using the BY variable, then use the "FIRST."/"LAST." Expressions to test for duplicates. If it is the 1st and last variable, drop it, otherwise list the duplicates.

```
proc sort;
    by Full_Name;
run;
data dupname;
    set vfw6796mailinglist;
    by Full_Name;
    if full_name ne ' ';
    if first.Full_Name and last.Full_Name then delete;
        if last.Full_Name then output; else delete;
run;
title3 'Duplicated Names from Mailing List';
proc print n u;
    var Full Name
        Delivery_Address
        City Zip4;
run;
```

The output listing looked like this, containing only the duplicated records by name.

Duplicated Names from Mailing List								
Obs	Full_Name		Delivery_Address	City	ZIP4			
1	Brenda Kilpatrick		11720 Red Bud Ln	Balch Springs	75180-21þ5			
2	Clyde Russell		3417 Hightrail Ln	Garland Garland	75043-1414			
3	David Kindle		6225 Goliad Ave	DALLAS	75214-3633			
4	Edna Geer		2716 Biloxi Ln	Mesquite	75150-1116			
5	Gary A Davis		1514 Larkspur Dr	ARLINGTON	76013-3671			
6	Gary L Cleveland		816 Courtenay Pl	Garland Garland	75040-5929			
7	Ida Caddel		PO Box 717	Omaha	75571-0717			
8	James G	Anderson	9941 Edgecliff Dr	DALLAS	75238-2645			
9	James W Anderson		5730 Sky Country St	San Antonio	78247-1440			
10	John E	Crady	2370 Norwood Dr	DALLAS	75228-2833			
11	John Tymensky		2203 Glengariff Dr	DALLAS	75228-2929			
12	Joseph R Yanes		5118 Matagorda Bay Ct	Rowlett	75089-4141			
13	Louis J Gillespie		624 Fieldwood Dr	Mesquite	75150-4905			
14	Melvin R	ELLIS	10120 Antelope Way	Forney	75126-7878			
15	Randall W Balliett		18135 Crestline Dr	Lake Oswego	97034-6220			
16	Ray Cook		9590 Forest Ln Apt 401	DALLAS	75243-5957			
17	Steve A	Haney	2946 N Eagle Nest Ave	Odessa	79764-9170			
18	Thomas Maldonado		3867 Dunhaven Rd	DALLAS	75220-3733			
19	Virginia Kenyon		318 Rusk St	Pittsburg	75686-1632			
20	Walter P Ditto		4601 Carlton Dr	Garland	75043-2165			
			N = 20					

So I have only 20 duplicate names. But those members that have spouses that are also members get two copies, so, to find those addresses that get multiple copies, sort by delivery address and check for duplicates.

VFW POST 67	96 Vet Gazette Newsletter Mailing	List Cleanup R	•	42
			12:47 Wednesday,	September 1, 2010
	Duplicate Addresses - Possible	Spouse		
Delivery_Address	Full_Name	City	ZIP4	
8354 Stony Creek Dr	Virginia A Stegall	DALLAS	75228-5817	
838 Brookshire Cir	Anne Sharon Perry	Garland	75043-5015	
838 Brookshire Cir	Katherine Moreno	Garland	75043-5015	
9418 Mercer Dr	Christopher J Worden	DALLAS	75228-4144	
9418 Mercer Dr	Julie Worden	DALLAS	75228-4144	
9941 Edgecliff Dr	James G Anderson	DALLAS	75238-2645	
9941 Edgecliff Dr	James G Anderson	DALLAS	75238-2645	
9941 Edgecliff Dr	James M Anderson Jr.	DALLAS	75238-2645	
	N = 113			

So sorting by address got 113 records that were duplicated. A small number but a lot of savings for this non-profit.

Conclusion

Finding duplicates is simple with SAS "FIRST." and "LAST." expressions. Find duplicates save resources, ie, money, that can be used for other tasks. Using the FIRST. And LAST. expressions is a quick and easy way to find duplicated data. Using SAS expressions can save a lot of coding time.

Author

Clarence Wm. Jackson, CSQA CJac.csqa@sbcglobal.net @CJac73 (twitter)