Paper 055-2009

LAG - the Very Powerful and Easily Misused SAS® Function

Yunchao (Susan) Tian, Social & Scientific Systems, Inc., Silver Spring, MD

ABSTRACT

There are times when SAS® programmers need to relate the value of a variable in the current observation to the value of the same or another variable in the previous observation. The LAG function is a very useful tool for this purpose. Unfortunately, it is frequently misused and produces unexpected results. This is because it often appears that the LAG function returns the value of the variable from the previous observation, which is true if it is called in non-conditional code. However, when it is executed conditionally, the LAG function only retrieves values from observations for which the condition is satisfied.

This paper discusses how the LAG function should be used in different situations to avoid unexpected results. The intended audience is beginner to intermediate SAS users with good knowledge of Base SAS.

INTRODUCTION

The LAG function is one of the techniques for performing computations across observations. A LAGn (n=1-100) function returns the value of the nth previous execution of the function. It is easy to assume that the LAGn functions return values of the nth previous observation. This is true if you process the data file sequentially. However, if LAG is called in a conditional statement, it actually returns values from previous calls, which may not be previous observations. So it is important to know that the LAG function should not be executed conditionally. First, execute the LAG function and assign the results to a new variable, then use the new variable for the conditional processing. This is best illustrated by examples.

EXAMPLE 1: COMPUTE HOUSE PRICE INCREASES FROM YEAR TO YEAR FOR ONE COUNTY

Suppose you have a SAS data set called EXAMPLE1 which contains the median sales prices of houses sold from 2001 to 2005 for a county. Data set EXAMPLE1 has the following contents:

Data set EXAMPLE1

YEAR	PRICE
2001	200000
2002	220000
2003	250000
2004	280000
2005	310000

Now you want to calculate the percent increase in PRICE from each year to the next. The DATA step below uses the LAG function to accomplish this.

```
DATA EXAMPLE1_RIGHT;
  SET EXAMPLE1;
  LAG_PRICE = LAG(PRICE);
  DIF_PRICE = PRICE - LAG_PRICE;
  PER_INCREASE = (DIF_PRICE/LAG_PRICE)*100;
RUN;
```

The above code first calculates the price of the previous year (LAG_PRICE) and the price difference (DIF_PRICE) from one year to the next. Then the percent increase is calculated from these two values. The output from PROC PRINT is shown in Table 1.

Table 1. *Example 1: Compute House Price Increases from Year to Year for One County*

YEAR	PRICE	LAG_ PRICE	DIF_ PRICE	PER_ INCREASE
2001	200000	.	.	.
2002	220000	200000	20000	10.0
2003	250000	220000	30000	13.6
2004	280000	250000	30000	12.0
2005	310000	280000	30000	10.7

As expected, the first observation has missing values for all calculated variables. Here you can also use the DIF function to calculate the price difference. DIF works in the same way as LAG, except that it returns the difference between the value of the current observation and its lag, i.e. DIF(PRICE) = PRICE – LAG(PRICE). For illustration purpose, three statements are used to calculate the percent increase. But you actually only need the following one statement:

```
PER_INCREASE = (DIF(PRICE)/LAG(PRICE))*100;
```

In this example, the data file is processed sequentially, so the LAG function returns the value of the previous observation. Now let us see in the next example what kind of trouble you can get into if you use LAG or DIF in a conditional statement.

EXAMPLE 2: COMPUTE HOUSE PRICE INCREASES FROM YEAR TO YEAR FOR MULTIPLE COUNTIES

In this example, the data contain house prices for three counties and we want to calculate the percent increase for every county. To demonstrate how to do this, we use the following SAS data set called EXAMPLE2 already sorted by COUNTY and YEAR:

Data set EXAMPLE2

COUNTY	YEAR	PRICE
1001	2001	200000
1001	2002	220000
1001	2003	250000
1001	2004	280000
1001	2005	310000
1002	2001	220000
1002	2002	240000
1002	2003	270000
1002	2004	300000
1002	2005	340000
1003	2001	280000
1003	2002	300000
1003	2003	330000
1003	2004	370000
1003	2005	410000

Here you can't process the data set sequentially since it is meaningless to compare the price of 2005 of one county with the price of 2001 of another county. You should only calculate the price increases within one county. Having this in mind, the following DATA step is the first attempt to do this:

```
DATA EXAMPLE2_WRONG;
  SET EXAMPLE2;
  BY COUNTY YEAR;
  IF NOT FIRST.COUNTY THEN DO;
    LAG_PRICE = LAG(PRICE);
    DIF_PRICE = DIF(PRICE);
    PER_INCREASE = (DIF_PRICE/LAG_PRICE)*100;
  END;
RUN;
```

Again, for illustration purpose, the above code used three statements instead of one statement to calculate the percent increase. The PROC PRINT output from this code is given in Table 2a.

Table 2a. *Example 2: Compute House Price Increases from Year to Year for Three Counties Wrong Result*

Obs	COUNTY	YEAR	PRICE	LAG_ PRICE	DIF_ PRICE	PER_ INCREASE
1	1001	2001	200000	.	.	.
2	1001	2002	220000	.	.	.
3	1001	2003	250000	220000	30000	13.6
4	1001	2004	280000	250000	30000	12.0
5	1001	2005	310000	280000	30000	10.7
6	1002	2001	220000	.	.	.
7	1002	2002	240000	310000	-70000	-23
8	1002	2003	270000	240000	30000	12.5
9	1002	2004	300000	270000	30000	11.1
10	1002	2005	340000	300000	40000	13.3
11	1003	2001	280000	.	.	.
12	1003	2002	300000	340000	-40000	-12
13	1003	2003	330000	300000	30000	10.0
14	1003	2004	370000	330000	40000	12.1
15	1003	2005	410000	370000	40000	10.8

Obviously this is not what we expected. Take observation 12 as an example, the value of LAG_PRICE is the value of PRICE of observation 10, not observation 11. Since LAG is called in conditional code, the value it returned is not the value from the previous observation, but is the value of its previous execution. To correct this problem, move the LAG and DIF functions outside of the conditional code to let them execute in every observation, and then set the values of LAG_PRICE and DIF_PRICE to missing for the first year of each county before calculating the percent increase, as in the following DATA step:

```
DATA EXAMPLE2_RIGHT;
  SET EXAMPLE2;
  BY COUNTY YEAR;
  LAG_PRICE = LAG(PRICE);
  DIF_PRICE = DIF(PRICE);
  IF FIRST.COUNTY THEN DO;
    LAG_PRICE = .;
    DIF_PRICE = .;
  END;
  PER_INCREASE = (DIF_PRICE/LAG_PRICE)*100;
RUN;
```

The corrected results are shown in Table 2b.

Table 2b. *Example 2: Compute House Price Increases from Year to Year for Three Counties Right Result*

Obs	COUNTY	YEAR	PRICE	LAG_ PRICE	DIF_ PRICE	PER_ INCREASE
1	1001	2001	200000	.	.	.
2	1001	2002	220000	200000	20000	10.0
3	1001	2003	250000	220000	30000	13.6
4	1001	2004	280000	250000	30000	12.0
5	1001	2005	310000	280000	30000	10.7
6	1002	2001	220000	.	.	.
7	1002	2002	240000	220000	20000	9.1
8	1002	2003	270000	240000	30000	12.5
9	1002	2004	300000	270000	30000	11.1
10	1002	2005	340000	300000	40000	13.3
11	1003	2001	280000	.	.	.
12	1003	2002	300000	280000	20000	7.1
13	1003	2003	330000	300000	30000	10.0
14	1003	2004	370000	330000	40000	12.1
15	1003	2005	410000	370000	40000	10.8

EXAMPLE 3: IMPUTE MISSING PRICE BASED ON PREVIOUS YEAR'S PRICE

In reality, data are much more complicated and usually have missing values. Very often missing values need to be imputed prior to analysis. In this example, we want to impute the missing price based on a 10% increase from the price of the previous year or a 20% increase from the price of two years ago if the price of the previous year is also missing. We will use the following data set EXAMPLE3 to demonstrate how to accomplish this.

Data set EXAMPLE3

COUNTY	YEAR	PRICE
1001	2001	200000
1001	2002	.
1001	2003	.
1001	2004	280000
1001	2005	310000
1002	2001	.
1002	2002	240000
1002	2003	270000
1002	2004	300000
1002	2005	340000
1003	2001	280000
1003	2002	300000
1003	2003	330000
1003	2004	370000
1003	2005	.

First, let's keep things simple and do the imputation for just one county. The intent of the following DATA step is to impute the missing price of 2005 for the last county.

```
DATA EXAMPLE3_WRONG;
  SET EXAMPLE3 (WHERE=(COUNTY=1003));
  IF PRICE NE . THEN PRICE_IMPUTE = PRICE;
  ELSE PRICE_IMPUTE = LAG(PRICE)*1.1;
RUN;
```

The results of this DATA step are shown in Table 3a.

Table 3a. *Example 3: Impute Missing Price Based on Previous Year's Price*
Wrong Result

COUNTY	YEAR	PRICE	PRICE_ IMPUTE
1003	2001	280000	280000
1003	2002	300000	300000
1003	2003	330000	330000
1003	2004	370000	370000
1003	2005	.	.

As you can see, the missing price is not imputed. What happened here is that the LAG function is called only when PRICE has missing value. As a result only the missing value goes in and comes out. Again, to obtain the expected results, we have to move LAG outside of the conditional statement and proceed as follows:

```
DATA EXAMPLE3_RIGHT (DROP=LAG_PRICE LAG2_PRICE LAG_COUNTY LAG2_COUNTY);
SET EXAMPLE3;
  LAG_PRICE = LAG(PRICE);
  LAG2_PRICE = LAG2(PRICE);
  LAG_COUNTY = LAG(COUNTY);
  LAG2_COUNTY = LAG2(COUNTY);
  IF PRICE NE . THEN PRICE_IMPUTE = PRICE;
  ELSE IF PRICE = . AND COUNTY = LAG_COUNTY AND LAG_PRICE NE .
    THEN PRICE_IMPUTE = LAG_PRICE*1.1;
  ELSE IF PRICE = . AND LAG_PRICE = . AND COUNTY = LAG2_COUNTY
    AND LAG2_PRICE NE . THEN PRICE_IMPUTE = LAG2_PRICE*1.2;
RUN;
```

The results from the above code are given in Table 3b. All missing prices are imputed except for the one of 2001 which can't be imputed since there is no previous year's price to be used for calculation. Here the LAG2 function is used when there are missing prices for two consecutive years. The LAG2 – LAG100 functions work in similar manner as the LAG function. The LAG function lags by one and the LAG2 function lags by two, and so on.

Table 3b. *Example 3: Impute Missing Price Based on Previous Year's Price Right Result*

Obs	COUNTY	YEAR	PRICE	PRICE_ IMPUTE
1	1001	2001	200000	200000
2	1001	2002	.	220000
3	1001	2003	.	240000
4	1001	2004	280000	280000
5	1001	2005	310000	310000
6	1002	2001	.	.
7	1002	2002	240000	240000
8	1002	2003	270000	270000
9	1002	2004	300000	300000
10	1002	2005	340000	340000
11	1003	2001	280000	280000
12	1003	2002	300000	300000
13	1003	2003	330000	330000
14	1003	2004	370000	370000
15	1003	2005	.	407000

Of course, we can have a different imputation algorithm to impute the missing price of one year from the price of next year by sorting the data in descending order of year. We can also impute missing prices in three or more consecutive years by using LAG3, etc.

CONCLUSION

We have examined the LAG function in depth. It does require a good understanding of this tool in order to use it properly. Always remember to first take the LAG of a specific variable and then conditionally use it when needed. If you try to conditionally use the LAG function you will get unexpected results and an error of this kind may not even be detected. This paper is by no means an exhaustive collection of uses of the LAG function, but it should give you a good start on using it as one of the techniques for performing operations across observations.

ACKNOWLEDGMENTS

I would like to thank my project manager Arlene Turner for her support and my colleague Dr. Raymond Hu for helpful discussions.

CONTACT INFORMATION

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

Yunchao (Susan) Tian
Social & Scientific Systems, Inc.
8757 Georgia Avenue, 12th Floor
Silver Spring, MD 20910
Work Phone: (301) 628-3285
Fax: (301) 628-3201
Email: Stian@s-3.com