

LAGging Questions

Debbie Hoppe, John Alden Life Insurance Company, Sacramento, CA

Introduction

The LAG function in SAS is a powerful but tricky tool. The entire key to understanding how the LAG function works can be summed up as:

LAG stores and returns values when it is executed.

The purpose of this paper is to illustrate how LAG stores values, when it stores values, and when it returns values.

Syntax

The syntax for using the LAG function is:

LAG<n>(argument)

where:

n=the number of lagged values. A LAG statement without an n value defaults to 1.

argument=any numeric or character value, typically a variable name (but it doesn't have to be.)

For example the SAS statement,

lastmo=lag(thismo);

would return the value of the variable *thismo* from the previous invocation of the LAG function, while the SAS statement

two_ago=lag2(thismo);

would return the value of the variable *thismo* from the second previous invocation of the LAG function.

Building the LAG Queue

LAG works by building queues which store and return values. Each occurrence of LAG function in a data step builds its own queue.

The queue for the LAGn function is initialized with n missing values. For example, for the LAG4 function, a LAG queue with 4 missing values would be created.

Lag stores and returns values when it is executed.

When the LAG function is executed, the value at the top of the queue is removed and returned, and the remaining values are shifted upwards. The new value of the argument is placed at the bottom of the queue. So for the first n executions of a LAG function, a missing value will be returned.

This simple example shows how the values are returned.

```
DATA LAGDEMO;
INPUT @1 THISMO DATE7.;
FORMAT THISMO DATE7. LASTMO DATE7.
       TWO_AGO DATE7.;
LASTMO=LAG(THISMO);
TWO_AGO=LAG2(THISMO);
CARDS;
01JAN95
25JAN95
04FEB95
08MAR95
10APR95
28MAY95
11JUN95
15JUL95
;;;
PROC PRINT;
```

produces this output.

OBS	THISMO	LASTMO	TWO_AGO
1	01JAN95	.	.
2	25JAN95	01JAN95	.
3	04FEB95	25JAN95	01JAN95
4	08MAR95	04FEB95	25JAN95
5	10APR95	08MAR95	04FEB95
6	28MAY95	10APR95	08MAR95
7	11JUN95	28MAY95	10APR95
8	15JUL95	11JUN95	28MAY95

The building and retrieval of the values from the queues is very straightforward - as long as there is no conditional execution. But -

LAG stores and returns values when it is executed.

So, let's throw some conditional execution into the program and see what happens.

```
DATA LAGDEMO;
INPUT @1 THISMO DATE7.;
FORMAT THISMO DATE7. LASTMO DATE7.
       TWO_AGO DATE7.;
IF THISMO GT '01MAR95'D THEN LASTMO=LAG(THISMO);
IF THISMO GT '01FEB95'D THEN
TWO_AGO=LAG2(THISMO);
CARDS;
01JAN95
15JUL95
25JAN95
08MAR95
28MAY95
04FEB95
11JUN95
10APR95
;;;
PROC PRINT;
```

produces this output.

OBS	THISMO	LASTMO	TWO_AGO
1	01JAN95	.	.
2	15JUL95	.	.
3	25JAN95	.	.
4	08MAR95	15JUL95	.
5	28MAY95	08MAR95	15JUL95
6	04FEB95	.	08MAR95
7	11JUN95	28MAY95	28MAY95
8	10APR95	11JUN95	04FEB95

Using any kind of subsetting or conditional logic in conjunction with LAG processing needs to be approached very carefully. These two programs contain the same statements, but because of the differences in execution, produce very different results.

The Moral of the Story

LAG is one of the few ways in SAS that you can peek back into the data in previous observations. Used with some understanding and care, it is a powerful and helpful tool. But you have to remember

LAG stores and returns data when it is executed.

```
DATA LAGDEMO;
INPUT @1 THISMO DATE7.;
FORMAT THISMO DATE7. LASTMO DATE7.
      TWO_AGO DATE7.;
LASTMO=LAG(THISMO);
TWO_AGO=LAG2(THISMO);
IF THISMO GT '01FEB95'D;
CARDS;
01JAN95
15JUL95
25JAN95
08MAR95
28MAY95
04FEB95
11JUN95
10APR95
;;;
PROC PRINT;
```

produces this output.

OBS	THISMO	LASTMO	TWO_AGO
1	15JUL95	01JAN95	.
2	08MAR95	25JAN95	15JUL95
3	28MAY95	08MAR95	25JAN95
4	04FEB95	28MAY95	08MAR95
5	11JUN95	04FEB95	28MAY95
6	10APR95	11JUN95	04FEB95

while if you move the subsetting if-statement in the program like this

```
DATA LAGDEMO;
INPUT @1 THISMO DATE7.;
FORMAT THISMO DATE7. LASTMO DATE7.
      TWO_AGO DATE7.;
IF THISMO GT '01FEB95'D;
LASTMO=LAG(THISMO);
TWO_AGO=LAG2(THISMO);
CARDS;
01JAN95
15JUL95
25JAN95
08MAR95
28MAY95
04FEB95
11JUN95
10APR95
;;;
PROC PRINT;
```

this is the result.

OBS	THISMO	LASTMO	TWO_AGO
1	15JUL95	.	.
2	08MAR95	15JUL95	.
3	28MAY95	08MAR95	15JUL95
4	04FEB95	28MAY95	08MAR95
5	11JUN95	04FEB95	28MAY95
6	10APR95	11JUN95	04FEB95